



# VOICE SERVER HTTP GATEWAY

*Versão 1.0*

Esta API permite o acesso a todas as funcionalidades do 360nrs usando JSON.

# ÚLTIMAS ALTERAÇÕES

Versão 1.0 15/02/2018 Versão Inicial

Versão 1.4 16/07/2018 Exemplos de Python, Java e C# adicionados

# ÍNDICE

INTRODUÇÃO	Pág. 4
PLATAFORMA TÉCNICA	Pág. 4
Pedido de envio de Voz	Pág. 5
Parâmetros	Pág. 5
Exemplo de pedido básico	Pág. 6
Exemplo de pedido CURL	Pág. 6
Exemplo de pedido PHP	Pág. 6
Exemplo de pedido PYTHON	Pág. 7
Exemplo de pedido JAVA	Pág. 8
Exemplo de pedido C#	Pág. 9
Códigos de estado de resposta	Pág. 10
Exemplos de resposta	Pág. 11
Política de novas tentativas	Pág. 11

## INTRODUÇÃO

A plataforma REST Gateway permite ao utilizador enviar mensagens de voz via HTTP ou HTTPS de forma simples e rápida, podendo enviar mais de 500 mensagens num único pedido. Para poder aceder às suas estatísticas e informações de cobrança pode aceder a <https://www.360nrs.com> com os dados de utilizador.

A documentação seguinte descreve os parâmetros necessários para utilizar todas as possibilidades de envio de mensagens de voz de acordo com as especificações REST. Quer os pedidos como as respostas da API REST estão em formato JSON, facilitando bastante o uso API com qualquer linguagem de programação.

## PLATAFORMA TÉCNICA

Cada pedido efetuado deverá incluir no respetivo cabeçalho do pedido http a autenticação do cliente. Tal é feito usando uma autenticação de acesso básico HTTP.

O cabeçalho da autorização é construído combinando a combinação "utilizador:palavra-passe" e codificando-a em base64. A esta combinação antepõe-se "Authorization: Basic"

Por exemplo, para o utilizador "miuser" e palavra-passe "mipass" o cabeçalho resultante seria: Authorization: Basic bWl1c2VyOm1pcGFzcw==

Em seguida, são exibidas as opções de envio disponíveis, o URL que se deve utilizar e os parâmetros admitidos.

## ENVIO DE PEDIDO DE VOZ

URL: <https://dashboard.360nrs.com/api/rest/voice>

MÉTODO: **POST**

### PARÂMETROS

Parâmetro	Tipo	Obrigatório	Descrição
message	string	Sim	Corpo da mensagem. Pode conter no máximo 500 caracteres. O texto deve ser codificado em UTF-8
to	array	Sim	Número do telemóvel destinatário da mensagem. Deve incluir o prefixo (Ex.: em Espanha 34666666666). Este campo permite-lhe especificar vários destinatários.
language	string	Sim	Idioma para converter o texto, as opções são: 'en_GB' => 'Inglês – Reino Unido', 'en_US' => 'Inglês – Estados unidos', 'es_CO' => 'Espanhol- Latino', 'es_ES' => 'Espanhol - Espanha', 'pt_BR' => 'Português - Brasil'
gender	string	Sim	Género da voz, os valores permitidos são "F" para voz de mulher e "M" para voz de homem.
scheduleDate	string	Não	Data de envio da mensagem em formato UTC. Caso seja necessário enviar mensagens programadas é possível especificar a data de envio indicando a data no formato YYYYmmddHHiiss (Exemplo: 20130215142000 seria 15 de fevereiro de 2013 às 14:20:00). Em caso de envio imediato não é necessário especificar este parâmetro.
expirationDate	string	Não	Permite determinar a data máxima na qual a chamada pode ser realizada. Após este prazo, a

			chamada será cancelada. Especificar o período de validade ativará automaticamente as tentativas de chamada, voltando a realizá-la caso o destinatário não atenda. A data deve ter o formato YYYYmmddHHiiSS (Exemplo: 20130215142000 seria 15 de fevereiro de 2013 às 14:20:00)
campaignName	String	Não	Nome da campanha. Caso seja especificado, será criada uma campanha com o nome indicado no painel de controle, contendo as estatísticas do envio. Caso já exista uma campanha com este nome, as estatísticas de envio serão juntas à campanha existente.
tags	array	Não	<i>campaignName é necessário caso este parâmetro seja especificado. Lista de tags a juntar à campanha. As tags podem ser utilizadas para filtrar as estatísticas no painel do controle.</i>

## EXEMPLO DE PEDIDO BÁSICO

```
{"to":["346666666666"],"message":"Este é um teste de mensagem de voz","gender":"F","language":"pt_BR"}
```

## EJEMPLO DE PETICIÓN CURL

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Authorization: Basic bWl1c2VyOm1pcGFzcw==" \
-d '{"to":["346666666666"],"message":"mensagem de texto","language":"pt_BR","gender":"M"}' \
https://dashboard.360nrs.com/api/rest/voice
```

## EXEMPLO DE PEDIDO PHP

```
<?php
$post["to"] = array("346666666666");
$post["message"] = "Este é um teste de mensagem de voz";
$post["gender"] = "F";
$post["language"] = "pt_BR";
$user = "miuser";
$password = "mipass";
```

```

try {
    $ch      = curl_init();
    curl_setopt($ch, CURLOPT_URL,
"https://dashboard.360nrs.com/api/rest/voice");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($post));
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($ch, CURLOPT_HTTPHEADER,
        array(
            "Accept: application/json",
            "Authorization: Basic ".base64_encode($user.":".$password));
    $result = curl_exec($ch);
    var_dump($result);
} catch (Exception $exc) {
    echo $exc->getTraceAsString();
}

```

## EXEMPLO DE PEDIDO PYTHON

```

import base64
import json
import pycurl
if __name__ == "__main__":

    url = "https://dashboard.360nrs.com/api/rest/voice"
    usrPass = "miuser:mipass"

    data = json.dumps({
        "to":["34666555444"],
        "message":"Este é um teste de mensagem de voz",
        "gender":"F",
        "language":"pt_BR",
        "campaignName":"Nome da Campanha",
    })

    b64Val = base64.b64encode(usrPass)
    headers = ["Accept:Application/json", "Authorization:Basic %s" % b64Val]

c = pycurl.Curl()
c.setopt(pycurl.URL, url)
c.setopt(pycurl.HTTPHEADER, headers)
c.setopt(pycurl.POST, 1)
c.setopt(pycurl.POSTFIELDS, data)
c.setopt(pycurl.SSL_VERIFYHOST, 0)
c.setopt(pycurl.SSL_VERIFYPEER, 0)
c.perform()

```

```
http_code = c.getinfo(pycurl.HTTP_CODE)
print(http_code)
```

## EXEMPLO DE PEDIDO JAVA

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;
import javax.net.ssl.HttpsURLConnection;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;

public class ApiVoice {

    public static void main(String args[]) throws IOException {
        String url = "https://dashboard.360nrs.com/api/rest/voice";
        URL obj = new URL(url);
        HttpsURLConnection con = (HttpsURLConnection)
obj.openConnection();

        con.setRequestMethod("POST");
        String userpass = "miuser" + ":" + "mipass";
        String basicAuth = "Basic " +
javax.xml.bind.DatatypeConverter.printBase64Binary(userpass.getBytes("UTF-8"));
        con.setRequestProperty("Authorization", basicAuth);
        con.setRequestProperty("Accept", "application/json");

        JSONObject data = new JSONObject();
        JSONArray to = new JSONArray();
        to.add("34666555444");
        data.put("to", to);
        data.put("message", "Este é um teste de mensagem de voz");
        data.put("gender", "F");
        data.put("language", "pt_BR");
        data.put("campaignName", "Nome da Campanha");
        String jsonText = data.toJSONString();

        con.setDoOutput(true);
        try (DataOutputStream wr = new
DataOutputStream(con.getOutputStream())) {
            wr.writeBytes(jsonText);
            wr.flush();
            wr.close();

        }

        BufferedReader in = new BufferedReader(
```



```

        new InputStreamReader(con.getInputStream());
        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();

        System.out.println(response.toString());
    }
}

```

## EXEMPLO DE PEDIDO C#

```

using Newtonsoft.Json;
using System;
using System.IO;
using System.Net;

namespace nrs_api
{
    class Program
    {
        static void Main(string[] args)
        {
            var httpRequest =
                (HttpWebRequest)WebRequest.Create("https://dashboard.360nrs.com/api/rest/voice");
            httpRequest.Method = "POST";
            httpRequest.Accept = "application/json";
            String username = "miuser";
            String password = "mipass";
            String encoded =
                System.Convert.ToBase64String(System.Text.Encoding.GetEncoding("ISO-8859-1").GetBytes(
                    username + ":" + password));
            httpRequest.Headers.Add("Authorization", "Basic " + encoded);

            using (var streamWriter = new
                StreamWriter(httpRequest.GetRequestStream()))
            {
                string[] to = new string[1];
                to[0] = "34666555444";
                var message = "Este é um teste de mensagem de voz";
                var gender = "F";
                var language = "pt_BR";
                var campaignName = "Nome da Campanha";

                var data = new

```

```

        {
            to = to,
            message = message,
            gender = gender,
            language = language,
            campaignName = campaignName,
        };
        string json = JsonConvert.SerializeObject(data);

        streamWriter.Write(json);
        streamWriter.Flush();
        streamWriter.Close();
    }

    var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
    using (var streamReader = new
StreamReader(httpResponse.GetResponseStream()))
    {
        var result = streamReader.ReadToEnd();
        Console.WriteLine(result);
        Console.ReadKey();
    }
}
}

```

A chave de acesso (palavra-passe) e o código do cliente (nome de utilizador) serão fornecidos pela empresa. É necessário referir que, para aumentar a segurança do sistema, o cliente deve indicar o IP a partir do qual ele se tenta ligar, pois apenas serão permitidos envios do IP indicado pelo cliente.

## CÓDIGOS DE ESTADO DA RESPOSTA

A API REST pode responder com os seguintes estados HTTP:

Código do estado	Descrição	Detalhes
202	Accepted	A mensagem foi aceite para processamento adicional
207	Multi-status	A mensagem foi aceite para processamento adicional, mas alguns dos destinatários estão incorretos.

400	Bad request	O pedido contém erros, a mensagem não foi aceite
401	Unauthorized	Falha na autenticação do cliente
402	Payment required	O cliente não possui saldo suficiente
500	Internal server error	Ocorreu um erro interno no servidor

## EXEMPLOS DE RESPOSTA

No corpo da resposta HTTP é entregue um JSON com as informações do resultado; as respostas possíveis são as seguintes:

CÓDIGO ESTADO 202 (ACCEPTED):

```
[{"accepted":true,"to":"34666555444","id":"102648819"}]
```

CÓDIGO ESTADO 207 (MULTI-STATUS):

```
[{"accepted":true,"to":"34626690739","id":"102648820"},{"accepted":false,"to":"34","error":{"code":102,"description":"No valid recipients"}}]
```

CÓDIGO ESTADO 400:

```
{"error":{"code":102,"description":"No valid recipients"}}
{"error":{"code":104,"description":"Text message missing"}}
{"error":{"code":105,"description":"Text message too long"}}
{"error":{"code":113,"description":"Invalid coding"}}
{"error":{"code":114,"description":"Invalid voice gender"}}
{"error":{"code":115,"description":"Invalid language"}}
{"error":{"code":116,"description":"Invalid validity period"}}
```

CÓDIGO ESTADO 401:

```
{"error":{"code":103,"description":"Username or password unknown"}}
{"error":{"code":112,"description":"IP address not allowed"}}
```

CÓDIGO ESTADO 402:

```
{"error":{"code":111,"description":"Not enough credits"}}
```

## POLÍTICA DE TENTATIVAS

O sistema de voz reage automaticamente às chamadas caso o destinatário não atender a chamada. Esta funcionalidade não gera custos acrescidos, uma vez que apenas as chamadas que sejam atendidas pelo destinatário serão cobradas. Quando o sistema deteta que a chamada não tenha sido atendida, tentará novamente passados 30 minutos. O tempo decorrido desde a chamada não atendida para a tentativa irá variar consoante a lista de espera existente, mas será sempre superior a 30 minutos. Serão realizadas um máximo de 3 tentativas após a primeira chamada. Para evitar que as novas tentativas sejam realizadas em horários indesejáveis é essencial especificar a variável opcional **expirationDate** de modo a ativar esta funcionalidade.