



# VOICE SERVER HTTP GATEWAY

*Versión 1.0*

Esta API permite el acceso a todas las características de 360nrs usando Json.

# ÚLTIMOS CAMBIOS

Versión 1.0 15/02/2018 Versión inicial

Versión 1.2 16/04/2018 Añadidos ejemplos Python, Java y C#

# ÍNDICE

INTRODUCCIÓN	Pág. 4
PLATAFORMA TÉCNICA	Pág. 4
Petición de envío de Voz	Pág. 5
Parámetros	Pág. 5
Ejemplo de petición básica	Pág. 6
Ejemplo de petición CURL	Pág. 6
Ejemplo de petición PHP	Pág. 6
Ejemplo de petición PYTHON	Pág. 7
Ejemplo de petición JAVA	Pág. 8
Ejemplo de petición C#	Pág. 9
Códigos de estado de respuesta	Pág. 10
Ejemplos de respuesta	Pág. 11
Política de reintentos	Pág. 11

## INTRODUCCIÓN

La plataforma REST Gateway permite al usuario enviar mensajes de voz a través de HTTP o HTTPS de una manera sencilla y rápida, pudiendo enviar más de 500 mensajes en una sola petición. Para poder acceder a sus estadísticas y datos de facturación puede acceder a la web <https://www.360nrs.com> con sus datos de usuario.

Esta documentación describe los parámetros necesarios para utilizar todas las posibilidades para el envío de mensajes de voz siguiendo las especificaciones REST. Tanto las peticiones como las respuestas de la API REST están en formato JSON, haciendo muy sencillo la utilización de la API con cualquier lenguaje de programación.

## PLATAFORMA TÉCNICA

Cada petición que se realice tendrá que incluir en la cabecera de la petición http la autenticación del cliente. Para ello se utiliza la autenticación de acceso básica de HTTP.

La cabecera de autorización se construye combinando la cadena “usuario:contraseña” y codificándola en base64. A esta cadena se antepone la cadena “Authorization: Basic”

Por ejemplo, para el usuario “miuser” y el password “mipass” la cabecera resultante sería:  
Authorization: Basic bWl1c2VyOm1pcGFzcmw==

A continuación se detallará las opciones de envío disponibles, la URL a la que se debe llamar, y los parámetros que admite.

## PETICIÓN ENVÍO DE VOZ

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

MÉTODO: **POST**

### PARÁMETROS

Parámetro	Tipo	Obligatorio	Descripción
message	string	Sí	Texto del mensaje. Como máximo puede tener 500 caracteres. El texto tiene que estar codificado en UTF-8
to	array	Sí	Número de teléfono móvil destinatario del mensaje. Debe incluir el prefijo (Ej: En España 34666666666). Este campo permite indicar multiples destinatarios.
language	string	Sí	Lenguaje para convertir el texto, las opciones son: 'en_GB' => 'Ingles – Reino Unido', 'en_US' => 'Ingles – Estados unidos', 'es_CO' => 'Español - Latino', 'es_ES' => 'Español - España', 'pt_BR' => 'Portugués - Brasil'
gender	string	Sí	Género de la voz, los valores permitidos son 'F' para voz de mujer y 'M' para voz de hombre.
scheduleDate	string	No	Fecha de envío del mensaje en formato UTC. Si se necesita enviar mensajes programados se puede especificar la fecha de envío indicando la fecha en formato YYYYmmddHHiiSS (Ej: 20130215142000 sería el 15 de febrero de 2013 a las 14:20:00). En caso de envío inmediato no se tiene que especificar este parámetro.
expirationDate	string	No	Permite determinar la fecha máxima en la que la llamada se puede realizar. Transcurrida esta fecha límite, la llamada será cancelada. Especificando el

			periodo de validez se activará automáticamente los reintentos de llamada, que volverá a realizar la llamada si el destinatario no atiende la llamada. La fecha debe tener el formato YYYYmmddHHii (Ej: 20130215142000 sería el 15 de febrero de 2013 a las 14:20)
campaignName	String	No	Nombre de campaña. Si se especifica se creará una campaña con el nombre indicado en el dashboard que contendrá las estadísticas del envío. Si una campaña con este nombre ya existe, las estadísticas de envío se añadirán a la campaña existente.
tags	array	No	<i>campaignName</i> es requerido si se especifica este parámetro. Listado de tags a añadir a la campaña. Los tags pueden ser utilizados para filtrar las estadísticas en el dashboard.

## EJEMPLO DE PETICIÓN BÁSICA

```
{"to":["346666666666"],"message":"Esto es un test de mensaje de voz","gender":"F","language":"es_ES"}
```

## EJEMPLO DE PETICIÓN CURL

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

## EJEMPLO DE PETICIÓN PHP

```
<?php

$post["to"] = array("346666666666");
$post["message"] = "Esto es un test de mensaje de voz";
$post["gender"] = "F";
$post["language"] = "es_ES";
$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();
}

```

## EJEMPLO DE PETICIÓN 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":"Esto es un test de mensaje de voz",
        "gender":"F",
        "language":"es_ES",
        "campaignName":"Nombre Campaña",
    })

    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)

```

## EJEMPLO DE PETICIÓN 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.HttpURLConnection;
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);
        HttpURLConnection con = (HttpURLConnection)
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", "Esto es un test de mensaje de voz");
        data.put("gender", "F");
        data.put("language", "es_ES");
        data.put("campaignName", "Nombre Campaña");
        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());
    }
}

```

## EJEMPLO DE PETICIÓN 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 = "Esto es un test de mensaje de voz";
                var gender = "F";
                var language = "es_ES";
                var campaignName = "Nombre Campaña";

                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();
}
}
}
}

```

La clave de acceso (password) y el código del cliente (username) serán proporcionados por la empresa. Hay que comentar que con objeto de aumentar la seguridad del sistema, el cliente deberá indicar la IP desde donde se va a conectar, solo se permitirán envíos de la IP indicada por el cliente.

## CÓDIGOS DE ESTADO DE RESPUESTA

La API REST puede responder con los estados HTTP siguientes:

Código estado	Descripción	Detalles
202	Accepted	El mensaje se ha aceptado para su posterior proceso
207	Multi-status	El mensaje se ha aceptado para su posterior proceso, pero algunos de los destinatarios son incorrectos.
400	Bad request	La petición contiene errores, el mensaje no ha sido aceptado
401	Unauthorized	Fallo en la autenticación del cliente

402	Payment required	El cliente no dispone de saldo suficiente
500	Internal server error	El servidor ha tenido un error interno

## EJEMPLOS DE RESPUESTA

En el cuerpo de la respuesta HTTP se entrega un JSON con los detalles del resultado, estas son las respuestas posibles:

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"}}
```

## PÓLITICA DE REINTENTOS

El sistema de voz permite reintentar automáticamente las llamadas si el destinatario no atiende la llamada. Esta funcionalidad no genera costes añadidos, ya que solo se cobrarán las llamadas que sean atendidas por el destinatario. Cuando el sistema detecta que la llamada no ha sido atendida, se encola nuevamente pasados 30 minutos. El tiempo transcurrido desde la llamada no atendida hasta el reintento será variable dependiendo de la cola existente, pero siempre superior a 30 minutos. Se reintentará un máximo de 3 veces después de la primera llamada. Para evitar que los reintentos se realicen en horario no deseados es imprescindible especificar la variable opcional **expirationDate** para que esta funcionalidad se active.