



PUSH APP SERVER  
HTTP GATEWAY  
*Version 1.2*

# Contents

<b>Introduction</b>	<b>2</b>
<b>Technical platform</b>	<b>2</b>
<b>Request to create PUSH APP campaign</b>	<b>3</b>
URL	3
Method	3
Parameters	3
Mandatory	3
name ( <i>string</i> )	3
type ( <i>string</i> )	3
groups ( <i>array</i> )	3
apps ( <i>array of arrays</i> )	3
os ( <i>array of arrays</i> )	4
Android	4
iOS	5
Optional	5
schedule ( <i>object</i> )	5
date ( <i>string</i> )	5
timezone ( <i>string</i> )	5
batch ( <i>object</i> )	6
parts ( <i>integer</i> )	6
minutes ( <i>integer</i> )	6
tags ( <i>array of strings</i> )	6
geolocation ( <i>object</i> )	6
latitude ( <i>float</i> )	6
longitude ( <i>float</i> )	6
radius ( <i>integer</i> )	6
segmentation ( <i>array of objects</i> )	7
field ( <i>string</i> )	7
condition ( <i>string</i> )	7
value ( <i>string</i> )	8
<b>Example of request PHP</b>	<b>9</b>
<b>Example of request PYTHON</b>	<b>10</b>
<b>Example of request JAVA</b>	<b>11</b>
<b>Example of request C#</b>	<b>13</b>
<b>Responses</b>	<b>16</b>
201 Successful request. Campaign created.	16
	1

422 Error in parameters.	16
400 Error creating campaign.	16

## Introduction

The http server platform allows users to create a campaign to send PUSH messages via this platform. To access your statistics and invoicing details, please go to <https://www.360nrs.com> and enter your user details.

Communication with the client will not be via any API provided by the company, as it will simply be HTTP communication, with some parameters to the indicated URL.

This process is detailed below.

## Technical platform

Each request made must include client authorisation in the header of the HTTP request. This will use basic HTTP access authentication.

The authorisation header is constructed by combining the "username: password" into a string and encoding it in Base64. This string is prefixed by the "Authorisation:Basic" string.

For example, for the user "miuser" and the password "mipass", the resulting header would be:

**Authorization: Basic bWl1c2VyOm1pcGFzcm==**

Detailed below are the available sending options, the URL that should be called, and the parameters supported.

# Request to create PUSH APP campaign

## URL

<https://dashboard.360nrs.com/api/rest/app-push>

## Method

POST

## Parameters

### Mandatory

name (*string*)

Campaign name

type (*string*)

Campaign type. Options:

- basic

groups (*array*)

Identifiers of the contact groups of the 360nrs platform to which you want to send the PUSH app campaign.

apps (*array of arrays*)

Identifiers of the apps registered on the 360nrs platform to which you want to send the PUSH app campaign.

os (*array of arrays*)

Composition details of the PUSH message for different operating systems (*Android and iOS*).

Depending on the type of notification chosen, some fields or others must be specified.

Android

### **notification\_type**

This is the field that will indicate the type of notification to send, so it is always mandatory. It is a string and the options are:

- basic
- advanced
- alertbox
- interstitial

Depending on the type of notification, the mandatory fields are the following:

#### **basic**

- title (*string*) (*maximum 30 characters*)
- subtitle (*string*) (*maximum 100 characters*)

#### **advanced**

- title (*string*) (*maximum 30 characters*)
- subtitle (*string*) (*maximum 100 characters*)

#### **alertbox**

- title (*string*) (*maximum 30 characters*)
- subtitle (*string*) (*maximum 100 characters*)
- buttonText (*string*) (*maximum 14 characters*)
- url (*string*)

#### **interstitial**

- image (*url or base64*)

iOS

### **notification\_type**

This is the field that will indicate the type of notification to send, so it is always mandatory. It is a string and the options are:

- basic
- advanced
- alertbox

Depending on the type of notification, the mandatory fields are the following:

#### **basic**

- title (*string*) (*maximum 30 characters*)
- subtitle (*string*) (*maximum 100 characters*)
- url (*string*)

#### **advanced**

- title (*string*) (*maximum 30 characters*)
- subtitle (*string*) (*maximum 100 characters*)

#### **alertbox**

- title (*string*) (*maximum 30 characters*)
- subtitle (*string*) (*maximum 100 characters*)
- buttonText (*string*) (*maximum 14 characters*)
- url (*string*)

## Optional

schedule (*object*)

Object with the send time at a future date:

date (*string*)

Send date with the format 'YYYY-mm-dd HH:ii:ss' (*MySQL*).

timezone (*string*)

Time zone to consider for send time.

E.g. "Europe/Madrid".

batch (*object*)

Object with the configuration of how many parts to be used and how often to send (in minutes).

parts (*integer*)

Number of parts into which you want to divide the batch.

minutes (*integer*)

How often (in minutes) from the start date you want to send the parts of the batch.

tags (*array of strings*)

Array of tags that you want to apply to the campaign.

e.g.

['blackfriday', 'sales']

geolocation (*object*)

Object with the geolocation settings of the campaign.

latitude (*float*)

Maximum latitude for sending messages. As it is a latitude, it must be between -90 and 90.

longitude (*float*)

Maximum longitude for sending messages. As it is a longitude, it must be between -90 and 90.

radius (*integer*)

Given the latitude and longitude point, the radius **in metres** to which you want the messages to arrive. It must be greater than 10 metres and less than 99999 metres.

segmentation (*array of objects*)

Array of conditions needed for the message to reach the user. Each item in the array is an object that has the following properties:

field (*string*)

Name of detail through which you want to segment. The following details are available:

- name
- surname
- gender
- age

condition (*string*)

Condition that the field has to meet to be delivered. The conditions available will depend on the chosen field.

- name
  - diff-to
  - equal-to
  - contains
  - start-with
  - end-with
- surname
  - diff-to
  - equal-to
  - contains
  - start-with
  - end-with
- gender
  - male
  - female
- age
  - diff-to
  - equal-to
  - greater-than
  - greater-than-or-equal-to
  - lower-than
  - lower-than-or-equal-to



value (*string*)

The value that the user data must have, comparing it using the condition sent in condition. Text comparisons do not take upper case into account (*case insensitive*).

E.g. (*javascript*)

```
{
  field: "name",
  condition: "equal-to",
  value: "Jose"
}
```

## Example of request Php

```
<?php
$post["name"]    = "Campaign name";
$post["type"]    = "basic";
$post["groups"]  = [2579];
$post["apps"]    = [
    "android" => [310]
];
$post["os"] = [
    "android" => [
        "notification_type" => "advanced",
        "title" => "title",
        "subtitle" => "subtitle",
        "image" => "https://www.w3schools.com/w3css/img_lights.jpg",
        "buttonText" => "button text an",
        "url" => "https://www.w3schools.com/w3css/img_lights.jpg",
        "options" => [
            "vibrate" => true,
            "sound" => true
        ]
    ],
    "ios" => [
        "notification_type" => "advanced",
        "title" => "title",
        "subtitle" => "subtitle",
        "image" => "https://www.w3schools.com/w3css/img_lights.jpg",
        "buttonText" => "button text ios",
        "url" => "https://www.apple.es",
        "options" => [
            "vibrate" => true,
            "sound" => true
        ]
    ]
];

$user      = "miuser";
$password  = "mipass";

try {
    $ch      = curl_init();
    curl_setopt($ch, CURLOPT_URL,
        "https://dashboard.360nrs.com/api/rest/app-push");
    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(
            "Content-type: application/json",
            "Authorization: Basic ".base64_encode($user.":".$password));
    $result = curl_exec($ch);
    var_dump($result);
} catch (Exception $exc) {
    echo $exc->getTraceAsString();
}

```

## Example of request Python

```

import pycurl
import base64
import json

if __name__ == "__main__":

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

    data=json.dumps({
        "name":"Campaign name",
        "type":"basic",
        "groups":[
            101
        ],
        "apps":{
            "android":[
                684,735
            ]
        },
        "os":{
            "android":{
                "notification_type": "advanced",
                "title": "title",
                "subtitle": "subtitle",
                "image": "https://www.w3schools.com/w3css/img_lights.jpg",
                "buttonText": "button text an",
                "url": "https://www.android.com",
                "options": {
                    "vibrate": True,
                    "sound": True
                }
            },
            "ios": {
                "notification_type": "advanced",
                "title": "title",
                "subtitle": "subtitle",
                "image": "https://www.w3schools.com/w3css/img_lights.jpg",
                "buttonText": "button text ios",
            }
        }
    })

```

```

        "url": "https://apple.es",
        "options": {
            "vibrate": True,
            "sound": True
        }
    }

    })
b64Val = base64.b64encode(usrPass)
headers=[
    "Content-type: 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)

```

## Example of request 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 ApiPushApp {

    public static void main(String args[]) throws IOException {
        String url = "https://dashboard.360nrs.com/api/rest/app-push";
        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("Content-type", "application/json");

        JSONObject data = new JSONObject();
        JSONArray groups = new JSONArray();
        JSONObject android = new JSONObject();
        JSONArray listAppsAndroid = new JSONArray();
        JSONObject notificationAndroid = new JSONObject();
        JSONObject notificationIos = new JSONObject();
        JSONObject osList = new JSONObject();
        JSONObject optionsAndroid = new JSONObject();
        JSONObject optionsIos = new JSONObject();

        data.put("name", "Campaign Name");
        data.put("type", "basic");

        groups.add(2579);
        data.put("groups", groups);

        listAppsAndroid.add(310);
        android.put("android", listAppsAndroid);
        data.put("apps", android);

        notificationAndroid.put("notification_type", "advanced");
        notificationAndroid.put("title", "title");
        notificationAndroid.put("subtitle", "subtitle");
        notificationAndroid.put("image",
"https://www.w3schools.com/w3css/img_lights.jpg");
        notificationAndroid.put("buttonText", "button text an");
        notificationAndroid.put("url", "https://www.android.com");
        optionsAndroid.put("vibrate", true);
        optionsAndroid.put("sound", true);
        notificationAndroid.put("options", optionsAndroid);

        notificationIos.put("notification_type", "advanced");
        notificationIos.put("title", "title");
        notificationIos.put("subtitle", "subtitle");
        notificationIos.put("image",
"https://www.w3schools.com/w3css/img_lights.jpg");
        notificationIos.put("buttonText", "button text ios");
        notificationIos.put("url", "https://apple.es");
        optionsIos.put("vibrate", true);
        optionsIos.put("sound", true);
        notificationIos.put("options", optionsIos);

        osList.put("android", notificationAndroid);
        osList.put("ios", notificationIos);

```

```

        data.put("os", osList);

        String jsonText = data.toJSONString();

        con.setDoOutput(true);
        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());

    }
}

```

## Example of request 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/app-
                push");
            httpRequest.Method = "POST";
            httpRequest.ContentType = "application/json";
            String username = "miuser";
            String password = "mipass";
            String encoded =
                System.Convert.ToBase64String(System.Text.Encoding.GetEncoding("ISO-8859-1").
                GetBytes(username + ":" + password));

```

```

        httpWebRequest.Headers.Add("Authorization", "Basic " + encoded);

        using (var streamWriter = new
StreamWriter(httpWebRequest.GetRequestStream()))
        {

            var name = "Campaign Name";
            var type = "basic";
            string[] groups = new string[1];
            groups[0] = "101";
            string[] appsAndroid = new string[1];
            appsAndroid[0] = "684";

            var apps = new
            {
                android = appsAndroid
            };

            var optionsAndroid = new
            {
                vibrate = true,
                sound = true
            };

            var notificationAndroid = new
            {
                notification_type = "advanced",
                title = "title",
                subtitle = "subtitle",
                image = "https://www.w3schools.com/w3css/img_lights.jpg",
                buttonText = "button text an",
                url = "https://www.android.com",
                options = optionsAndroid
            };

            var optionsIos = new
            {
                vibrate = true,
                sound = true
            };

            var notificarionIos = new
            {
                notification_type = "advanced",
                title = "title",
                subtitle = "subtitle",
                image = "https://www.w3schools.com/w3css/img_lights.jpg",
                buttonText = "button text ios",
                url = "https://www.apple.es",
                options = optionsIos
            };

```

```

        };

        var os = new
        {
            android = notificationAndroid,
            ios = notificarionIos
        };

        var data = new
        {
            name = name,
            type = type,
            groups = groups,
            apps = apps,
            os = os
        };

        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();
    }
}
}
}

```

## Responses

**201 Successful request. Campaign created.**

**422 Error in parameters.**

Specifying which parameter or parameters have failed in the response.



400 Error creating campaign.