

Question

[Yone Moreno](#) · Mar 5, 2020

Sending HTTP requests to store special characters

We would like to be able to store accented characters into an API.

Currently our Rest Operation is the following:

```
Class Operaciones.REST.NotificacionesPUSH.CrearNotificacion Extends EnsLib.REST.Operation
{ Parameter INVOCATION = "Queue"; Method CrearNotificacion(pRequest As
Mensajes.Request.NotificacionesPUSH.CrearNotificacion, pResponse As
Mensajes.Response.NotificacionesPUSH.CrearNotificacion) As %Library.Status
{
```

```
Set body = {
  "app_id": (pRequest.idApp),
  "headings": {"en":(pRequest.notificacion.titulo),"es":(pRequest.notificacion.titulo)},
  "subtitle": {"en":(pRequest.notificacion.subtitulo),"es":(pRequest.notificacion.subtitulo)},
  "contents": {"en":(pRequest.notificacion.mensaje),"es":(pRequest.notificacion.mensaje)},
  "data":(^data),
  "include_player_ids": (^idsDispositivos)
}
```

```
set httpRequest = ##class(%Net.HttpRequest).%New()
set tResponse = ##class(%Net.HttpResponse).%New()
set httpRequest.Server = "onesignal.com"
```

```
set httpRequest.ContentType = "application/json"
```

```
set httpRequest.Authorization = "Basic
" ##class(Util.TablasMaestras).getValorMaestra(
"NOTIFICACIONESPUSH.PARAMETRIZACIONES","ONESIGNAL_APIKEY_SCS")
```

```
set httpRequest.Https = 1 set httpRequest.SSLConfiguration = "Certificado_SCS"
```

```
set dataObject = ##class(%DynamicObject).%FromJSON(body.data)
```

```
set body.data = dataObject $$$LOGINFO("body.'include_player_ids': "_body.'include_player_ids")
set idsDispositivosObject = ##class(%DynamicObject).%FromJSON(body."include_player_ids")
```

```
set body."include_player_ids" = idsDispositivosObject //set status =
Do httpRequest.EntityBody.Write(body.%ToJSON()) while(httpRequest.EntityBody.AtEnd=0){
set linea = httpRequest.EntityBody.Read()
}
$$$LOGINFO("linea: "_linea)
```

```
set tSC = httpRequest.Post("https://onesignal.com/api/v1/notifications") set tResponse = httpRequest.
HttpResponse
if $$$ISERR(tSC){
$$$ThrowOnError(tSC)
}
if (tResponse.Data.AtEnd = 0) {

set linea = tResponse.Data.Read()

} set pResponse = ##class(Mensajes.Response.NotificacionesPUSH.CrearNotificacion).%New()
do pResponse.informacion.Write(linea) Quit pResponse
} XData MessageMap
{
<MapItems>
  <MapItem MessageType="Mensajes.Request.NotificacionesPUSH.CrearNotificacion">
    <Method>CrearNotificacion</Method>
  </MapItem>
</MapItems>
}}
```

We have read:

<https://community.intersystems.com/post/managing-utf-8-characters-databa...>

<https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls...>

<https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls...>

We have tried to use:

```
set httpRequest.ContentType = "application/json;charset=utf-8"
do httpRequest.SetHeader("Content-Type","application/json; charset=utf-8")
set httpRequest.ContentCharset = "utf-8"
```

And even we tried to convert the headings itself:

```
/"headings": {"en":($ZCONVERT(pRequest.notificacion.titulo,"O","UTF8")),"es":(pRequest.notificacion.titulo)},
/"headings": {"en":($ZCONVERT(pRequest.notificacion.titulo,"I","UTF8")),"es":(pRequest.notificacion.titulo)},
```

Currently we see that our operation's LOGINFORM let's us see that the JSON contains the accented character, in this case an "ó"

Texto:

```
linea: {"app_id": "-----", "headings": {"en": "Cita Atención Primaria", "es": "Cita Atención Primaria"}, "subtitle": {"en": "C.P.  
ISORA", "es": "C.P. ISORA"}, "contents": {"en": "Aquí el contenido del mensaje si aplicase", "es": "Aquí el contenido del mensaje si aplicase"}, "data": {"centro": "C.P.  
ISORA", "fecha": "yyy/mm/dd", "hora": "hh:mm", "profesional": "nombre del profesional", "nomUsuario": "nombre de  
usuario", "codcita": "idCita", "sepuedeborrar": "1"}, "include_player_ids": [{"-----"}]}
```

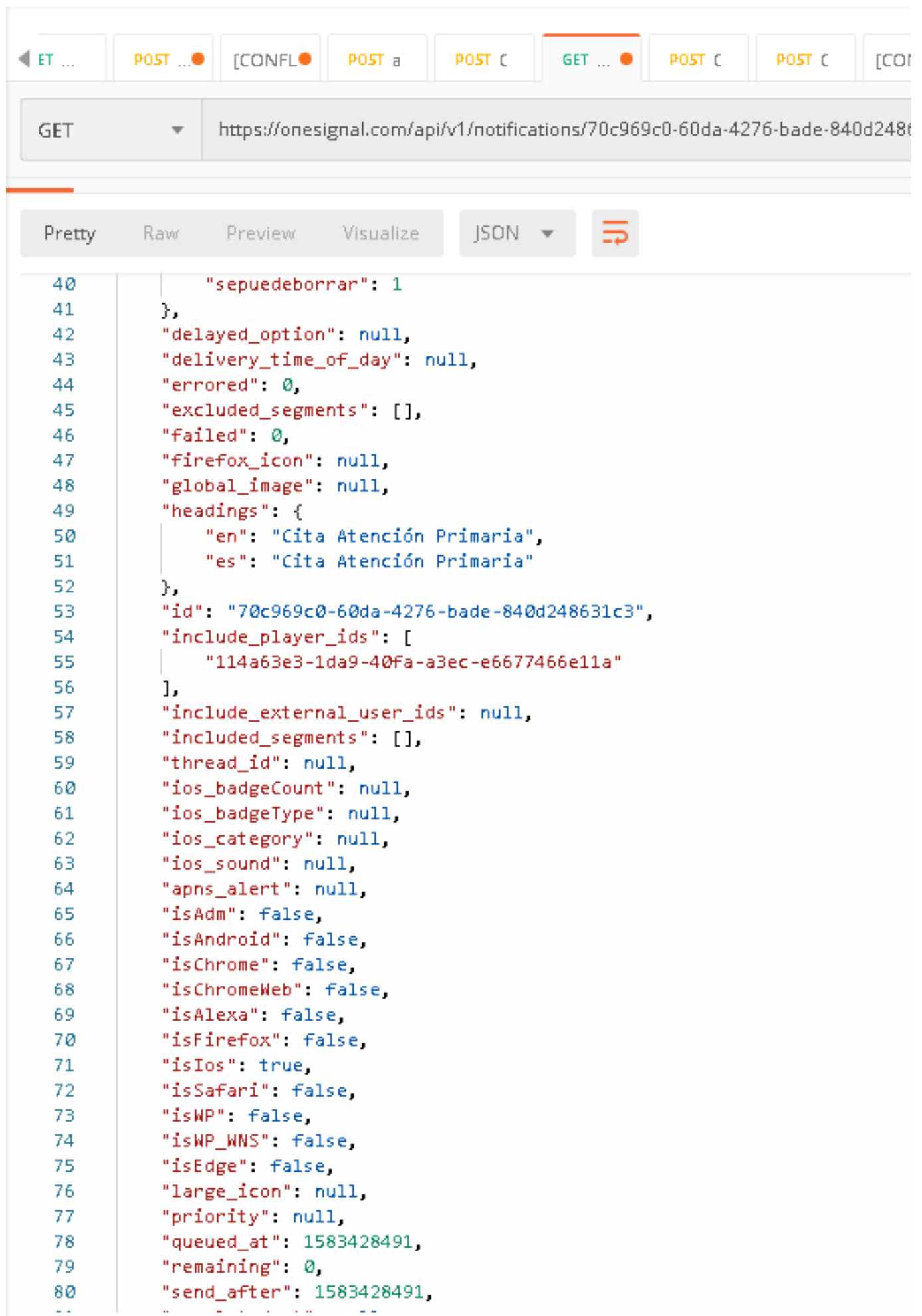
However when we view the notification stored in the API it shows as a strange character:

```
converted: 0,
"data": {
  "centro": "C.P. ISORA",
  "fecha": "yyy/mm/dd",
  "hora": "hh:mm",
  "profesional": "nombre del profesional",
  "nomUsuario": "nombre de usuario",
  "codcita": "idCita",
  "sepuedeborrar": "1"
},
"delayed_option": null,
"delivery_time_of_day": null,
"errored": 0,
"excluded_segments": [],
"failed": 0,
"firefox_icon": null,
"global_image": null,
"headings": {
  "en": "Cita Atención Primaria",
  "es": "Cita Atención Primaria"
},
"id": "9232f810-88e2-4771-acc6-d7d0fdccbf37",
"include_player_ids": [
  "114a63e3-1da9-40fa-a3ec-e6677466e11a"
],
"include_external_user_ids": null,
"included_segments": [],
"thread_id": null,
"ios_badgeCount": null,
"ios_badgeType": null,
"ios_category": null,
"ios_sound": null,
"apns_alert": null,
"isAdm": false,
"isAndroid": false,
"isChrome": false,
"isChromeWeb": false,
"isAlexa": false,
"isFirefox": false,
"isIos": true,
"isSafari": false,
"isWP": false.
```

Besides we have checked that if we send a create notification request directly from SoapUI, it is been stored with strange characters too:

```
converted: 0,
"data": {
  "centro": "C.P. ISORA",
  "fecha": "yyy/mm/dd",
  "hora": "hh:mm",
  "profesional": "nombre del profesional",
  "nomUsuario": "nombre de usuario",
  "codcita": "idCita",
  "sepuedeborrar": "1"
},
"delayed_option": null,
"delivery_time_of_day": null,
"errored": 0,
"excluded_segments": [],
"failed": 0,
"firefox_icon": null,
"global_image": null,
"headings": {
  "en": "Cita Atención Primaria",
  "es": "Cita Atención Primaria"
},
"id": "9232f810-88e2-4771-acc6-d7d0fdccbf37",
"include_player_ids": [
  "114a63e3-1da9-40fa-a3ec-e6677466e11a"
],
"include_external_user_ids": null,
"included_segments": [],
"thread_id": null,
"ios_badgeCount": null,
"ios_badgeType": null,
"ios_category": null,
"ios_sound": null,
"apns_alert": null,
"isAdm": false,
"isAndroid": false,
"isChrome": false,
"isChromeWeb": false,
"isAlexa": false,
"isFirefox": false,
"isIos": true,
"isSafari": false,
"isWP": false.
```

--> However if we set the Encoding to UTF-8 in SoapUI, it is being stored correctly:



The screenshot shows a REST client interface with a navigation bar at the top containing buttons for various HTTP methods: GET, POST, and CONFL. The main area displays a GET request to the URL `https://onesignal.com/api/v1/notifications/70c969c0-60da-4276-bade-840d248631c3`. Below the request, there are tabs for 'Pretty', 'Raw', 'Preview', and 'Visualize', and a dropdown menu set to 'JSON'. The response is displayed in a code editor with line numbers from 40 to 80. The JSON response is a notification object with various fields including `sepuedeborrar`, `delayed_option`, `delivery_time_of_day`, `errored`, `excluded_segments`, `failed`, `firefox_icon`, `global_image`, `headings` (with `en` and `es` values), `id`, `include_player_ids`, `include_external_user_ids`, `included_segments`, `thread_id`, `ios_badgeCount`, `ios_badgeType`, `ios_category`, `ios_sound`, `apns_alert`, `isAdm`, `isAndroid`, `isChrome`, `isChromeWeb`, `isAlexa`, `isFirefox`, `isIos`, `isSafari`, `isWP`, `isWP_WNS`, `isEdge`, `large_icon`, `priority`, `queued_at`, `remaining`, and `send_after`.

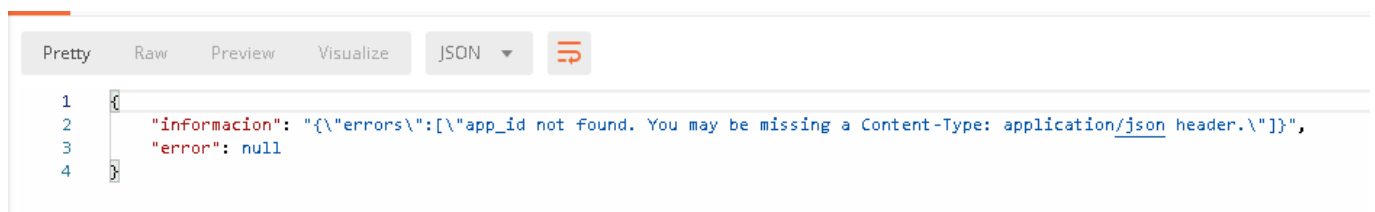
```
40     "sepuedeborrar": 1
41   },
42   "delayed_option": null,
43   "delivery_time_of_day": null,
44   "errored": 0,
45   "excluded_segments": [],
46   "failed": 0,
47   "firefox_icon": null,
48   "global_image": null,
49   "headings": {
50     "en": "Cita Atención Primaria",
51     "es": "Cita Atención Primaria"
52   },
53   "id": "70c969c0-60da-4276-bade-840d248631c3",
54   "include_player_ids": [
55     "114a63e3-1da9-40fa-a3ec-e6677466e11a"
56   ],
57   "include_external_user_ids": null,
58   "included_segments": [],
59   "thread_id": null,
60   "ios_badgeCount": null,
61   "ios_badgeType": null,
62   "ios_category": null,
63   "ios_sound": null,
64   "apns_alert": null,
65   "isAdm": false,
66   "isAndroid": false,
67   "isChrome": false,
68   "isChromeWeb": false,
69   "isAlexa": false,
70   "isFirefox": false,
71   "isIos": true,
72   "isSafari": false,
73   "isWP": false,
74   "isWP_WNS": false,
75   "isEdge": false,
76   "large_icon": null,
77   "priority": null,
78   "queued_at": 1583428491,
79   "remaining": 0,
80   "send_after": 1583428491,
--
```

So then, we would like to change the charset to utf-8 in Ensemble.

If we try:

```
do httpRequest.SetHeader("Content-Type","application/json; charset=utf-8")
```

The API does not recognize the Content-Type: application/json



The screenshot shows a JSON viewer interface with tabs for 'Pretty', 'Raw', 'Preview', and 'Visualize'. The 'JSON' tab is selected. The JSON content is as follows:

```
1 {
2   "informacion": "{\\"errors\\":[\\"app_id not found. You may be missing a Content-Type: application/json header.\\"]}",
3   "error": null
4 }
```

We have also read:

<https://stackoverflow.com/questions/17154967/is-content-encoding-being-s...>

How could we send HTTP requests to store special characters

[#ObjectScript #Atelier](#)

Source URL: <https://community.intersystems.com/post/sending-http-requests-store-special-characters%E2%80%BD>