

Telekom Push Notification Service - Server Interfaces

Table of Contents

1. Overview	1
1.1. Version information	1
1.2. Contact information	1
1.3. License information	1
2. Document History	1
3. Telekom Push Notification Service Introduction	2
3.1. Server URLs	2
3.2. Supported Application Types	2
3.3. Device Filters	3
4. Admin Interface	3
4.1. Device Registration	3
4.2. Device Unregistration	7
4.3. Retrieve Device List	8
5. Notification Interface	11
5.1. Diagnostic Headers	11
5.2. Send Push Notification	12
5.3. Send SMS	27
6. Phone Number Validation Interface	31
6.1. Phone Number Validation	31

1. Overview

This is the API guide for the Telekom Push Notification Service.

1.1. Version information

TPNS Release : 6.11.0

Release Date: 2022-10-05

1.2. Contact information

Contact : Team TPNS

Contact Email : tpns@telekom.de

1.3. License information

License : Apache 2.0

License URL : <http://www.apache.org/licenses/LICENSE-2.0>

Terms of service : urn:tos

2. Document History

TPNS Release	Release Date	Description
4.50.0	2019-07-03	Added property "channel_id" to AndroidNotification object
5.0.0	2019-07-31	Final version after switching to maintenance mode
5.3.1	2019-08-28	Added property "dataPush" and "title" to Send Push Notification Request Body
6.0.0	2020-06-23	No changes
6.1.0	2020-07-31	Huawei API documented
6.7.0	2022-01-19	Information about unsupported features removed.
6.9.0	2022-04-06	Notification interface docs extended.
6.10.0	2022-06-29	Added property "collapseKey" to Additional Parameters for iOS Devices .
6.10.5	2022-09-13	Added backend timer metrics (google, apple, sms).
6.10.6	2022-09-18	Added sender timer metrics (google, apple, sms).

3. Telekom Push Notification Service Introduction

The Telekom Push Notification Service (TPNS) API provides operations to manage devices and to send push notifications to registered devices. The TPNS API is a REST based service using [JSON](#) format to exchange data between clients and server. This requires the following **request headers to be included with every request** sent to the TPNS service:

Header Name	Header Value	Description
Content-Type	application/json	The Content-Type header defines the data format used in the request body. Only <i>application/json</i> is currently supported.
Accept	application/json	The Accept header defines the data format that is expected to be returned from the server to the client. Only <i>application/json</i> is currently supported.

3.1. Server URLs

The following URLs are used by TPNS:

Table 1. Server URLs

Environment	{SERVER_URL}
Test	https://tpns-preprod.molutions.de/TPNS/
Production	https://tpns.molutions.de/TPNS/

3.2. Supported Application Types

The following application types are currently supported by TPNS:

Table 2. Supported Application Types

Application Type	Description
ADM	Amazon Device Messaging
AOS	Android (all versions)
IOS	Apple production environment
IOS_SAND	Apple development environment

Application Type	Description
WOS10	Windows 10
SMS	SMS application
WEB	Web Browser Push
HMS	HUAWEI Mobile Services

3.3. Device Filters

The Telekom Push Notification Service (abbr. TPNS) enables the backend systems to broadcast messages to specific application devices. This is possible through adding additional parameters in the JSON structure of the HTTP request body.

The parameter 'filters' should contain a list of generic key-value pairs, which are used in filtering the devices that will receive the notification. If the value contains a % sign, a wildcard match is performed instead of an exact match.

Sample Request with exact and wildcard filter

```
[
  {
    "applicationKey": "sampleApp",
    "message": "This message will be sent to all devices",
    "applicationTypes": [...],
    "deviceIds": [...],
    "filters": [
      {
        "key": "hiRes",
        "value": "true"
      },
      {
        "key": "language",
        "value": "en%"
      }
    ]
  }
]
```

The notification will be sent to those devices that have all the filters saved as additional_parameters in the database.

If a filter isn't found in the additional_parameters field of a device, that device will not receive the notification. If no filter is defined in the list of filters, the notification will be sent to all devices specified by the other parameters in the push notification request.

4. Admin Interface

The admin interface provides operations to register a device, to unregister a previously registered device, and to query for a list of devices registered for an application.

4.1. Device Registration

Every mobile device (smartphone / tablet) can register by himself in order to receive notifications from an application using the TPNS. To register for the application's notification, the device has to send a request to the

TPNS device registration service, providing the ID of the application it is registering for, own device ID, OS type, and optional additional parameters that may be used as filters when sending notifications to the device.

The response sent back by TPNS will indicate if the registration could be completed successfully, or - in case of errors - provide a list of the invalid data elements sent by the device.

Any further request using the same device ID will overwrite previous registration. It means that no unregistration request is needed to update device's registration.

4.1.1. Request

The registration request has the following format:

```
POST {SERVER_URL}/api/device/register
```

4.1.2. Request Parameters

This service does not use any request parameters.

4.1.3. Request Body

The request body contains a JSON object as described in the following table.

Table 3. Device Registration Request Body

Property Name	Property Type	R/O	Description
deviceId	string(255)	R	The ID of the registering device
deviceRegistrationId	string(255)	R	The registration ID of the registering device; for Windows 10 the push channel URL assigned by WNS has to be provided instead
applicationKey	string(255)	R	The internal key of the application used to register the application in TPNS
applicationType	string(255)	R	The type of the OS (see Supported Application Types). Important: The registration will be rejected if no application type specific configuration has been created in the TPNS UI.
additionalParameters	array	O	Array of optional registration parameters to be used as filter criteria in notification requests. See table additionalParameters Array .

The *additionalParameters* array allows to specify application specific filter criteria that may be used to select devices when sending a push notification. Examples for application specific filter criteria are the user's customer number, the country of residence, the kind of device registering, and so on.

The *additionalParameters* is an array of key-value pairs as described below.

Table 4. additionalParameters Array

Property Name	Property Type	R/O	Description
key	string(255)	R	The name of the provided filter criteria.
value	string(500)	R	The value of the provided filter criteria.

Sample Request

```
{
  "applicationType": "IOS",
  "applicationKey": "appKey",
  "deviceId": "4379ad2f-65c6-3571-43e1-f0750eef75f6",
  "deviceRegistrationId": "47ECC80E1BA6047AE053EADDFB946112",
  "additionalParameters": [
    {
      "key": "location",
      "value": "darmstadt"
    },
    {
      "key": "propertyType",
      "value": "house"
    }
  ]
}
```

Sample Request for WOS10

```
{
  "applicationType": "WOS10",
  "applicationKey": "appKey",
  "deviceId": "862fc4bb-004b-d66c-be5f-161333f0b078",
  "deviceRegistrationId": "https://db5.notify.windows.com/?token=AwYAAAB3b..",
  "additionalParameters": [ ]
}
```

4.1.4. Response

Upon completion, an HTTP status of 200 will be sent back to the calling device, together with a response object in HTTP body indicating the registration result. The calling device has to check the response object to detect an unsuccessful registration attempt.

Table 5. Device Registration Response

HTTP Status	Description
200	The operation completed. The calling device has to check the response object to detect an unsuccessful registration attempt.
422	Unprocessable Entity. Request cannot be processed. ApplicationKey might be invalid. For possible Errors see table Device Possible Error Codes
500	Internal Server Error.

4.1.5. Response Body

Table 6. Device Registration Response

Property Name	Property Type	R/O	Description
success	boolean	R	indicates if the registration request is completed successfully ("true") or is failed ("false"). If the device registration failed, the <i>errors</i> element will contain a list of error messages.

Property Name	Property Type	R/O	Description
message	string	R	a message providing a general error description (e.g. internal server error). If a list of specific errors is provided in the <i>errors</i> element, this field may be empty.
errors	array	R	list of errors that were detected during device registration. Empty if the request completed successfully. See table errors Array
deviceId	string	R	The ID of the registered device. This is the ID specified in the registration request.

Table 7. errors Array

Property Name	Property Type	R/O	Description
field	string	R	the name of the field that caused the error
message	string	R	a message describing the error
errorCode	string	R	an error code for the occurred error. See table Device Possible Error Codes

Table 8. Device Possible Error Codes

Error Code	Description
BLACKLISTED	The device was blacklisted by the provider.
UNKNOWN_APPKEY	The application key is unknown.
INVALID_APPKEY	The application has no configuration for this backend type.

Sample Response (Success)

```
{
  "success":true,
  "message":"Device successfully updated",
  "errors":null,
  "deviceId":"862fc4bb-004b-d66c-be5f-161333f0b078"
}
```

Sample Response (Failure)

```
{
  "success": false,
  "message": "Registration Failed",
  "errors": [
    {
      "field": "deviceId",
      "message": "The device ID may not be missing/empty",
      "errorCode": "INVALID_DEVICE_ID"
    },
    {
      "field": "applicationType",
      "message": "Invalid application type ISO",
      "errorCode": "INVALID_APPLICATION_TYPE"
    }
  ]
}
```

4.2. Device Unregistration

Upon a user's request a device can unsubscribe from receiving push notifications from an application. To unsubscribe the device sends a request to the TPNS unregistration service providing the ID of the application and it's own device ID.

A device will be removed from the application's notification list by using of following request description.

4.2.1. Request

The unregistration request has the following format:

```
PUT
{SERVER_URL}/api/application/{application_key}/device/{device_id}/unregister
```

4.2.2. Request Parameters

Table 9. Device Unregistration Request Parameters

Property Name	Property Type	R/O	Description
application_key	string(255)	R	The ID of the application to unregister from.
device_id	string(255)	R	The ID of the registered device that wants to unregister.

4.2.3. Request Body

No request body is needed for this service.

4.2.4. Response

Upon successful completion, the HTTP status of 200 will be sent back to the calling device.

Table 10. Device Unregistration Response

HTTP Status	Description
200	The operation completed.

4.2.5. Response Body

Table 11. Device Unregistration Response Body

Property Name	Property Type	R/O	Description
success	boolean	O	Indicates if the request is completed successfully. The response body is only used upon success. If an error occurred the response body will be empty.
message	string	R	A message providing a general error description (e.g. internal server error). If a list of specific errors is provided in the <i>errors</i> element, this field may be empty.
errors	array	R	List of errors that were detected during device unregistration. Empty if the request completed successfully. See table errors Array

Table 12. errors Array

Property Name	Property Type	R/O	Description
field	string	R	The name of the field that caused the error
message	string	R	A specific error description
errorCode	string	R	An error code for the occurred error

Sample Response (Success)

```
{
  "success": true,
  "message": null,
  "errors": null
}
```

4.3. Retrieve Device List

Every backend system known in Telekom Push Notification Service is able to request the list of registered devices for a specific Application ID, with the use of a password. External backend systems can also retrieve the list of updates (registrations, unregistrations) for a particular period.

An external system could send an additional parameter to specify the exact number of minutes, in which TPNS will retrieve the modified list from the database. A call without this parameter will initialize a query for the last minute in the database.

4.3.1. Request

The device list request has the following format:

```
GET
{SERVER_URL}/api/application/{application_key}/devices?minutesPassed={minutes_passed}
POST
{SERVER_URL}/api/application/{application_key}/devices?minutesPassed={minutes_passed}
```

The POST request allows to provide additional filters in the request body (see below).

4.3.2. Request Parameters

Table 13. Device List Request Parameters

Property Name	Property Type	R/O	Description
application_key	string(255)	R	The ID of the application the list of registered devices is queried for.
minutes_passed	number	O	This parameter specifies the number of minutes to go back in time to search for new registrations/unregistrations. If this parameter is missing, TPNS will look for changes in the last minute only.

Table 14. Device List Headers

Header Name	Header Value	R/O	Description
api.pnserver.password.access	string(255)	R	The password of the application specified in the <i>application_key</i> parameter.

4.3.3. Request Body

The request body is only used with a POST request and allows to specify additional filter criteria to limit the number of devices returned. A JSON object with the following structure is expected:

Table 15. Device List Request Body

Property Name	Property Type	R/O	Description
status	string	O	If this value is omitted or blank, devices will be returned independent of their current registration status. A value of "active" only returns devices that currently have a valid registration. Any other value will be treated as "inactive" and thus return only devices that do not have a valid registration anymore.
additionalParams	array	O	Array of optional registration parameters to be used as filter criteria. The values provided here will be matched against the additional registration parameters provided during device registration. See table additionalParameters Array .

When sending a GET request the request body is not used.

Sample POST Request

```
{
  "status": "active",
  "additionalParams": [
    {
      "key": "accountId",
      "value": "9005-ac555dd5-9617-4556-83e6-ad11674a3374"
    }
  ]
}
```

4.3.4. Response

Upon successful completion, an HTTP status of 200 will be sent back to the calling device.

Table 16. Device List Response

HTTP Status	Description
200	The operation completed successfull.
403	The "api.pnserver.password.access" header is missing or the provided password is invalid. This value is also returned if the specified application is unknown.

4.3.5. Response Body

The response body contains a list of all devices registered for or unregistered from the respective application during the last minutes as specified in the *minutes_passed* parameter. The list will be empty if no devices have registered for unregistered during this time period. If the POST request has been used, the result list will only

contain devices that match the criteria provided in the request body.

Table 17. Device List Response Body

Property Name	Property Type	R/O	Description
deviceId	string	R	the ID of the device
applicationKey	string	R	The ID of the requesting application (same as <i>application_key</i> parameter).
applicationType	string	R	The application type of the application.
additionalParameters	string	R	A list of the additional registration parameters that were specified during the registration of the device. See table additionalParameters Array .
created	date	R	Date/time when the device was initially registered.
modified	date	R	Date/time when the device was last updated (re-registered or unregistered).
active	boolean	R	Indicates if the device is registered (true) or has been unregistered (false).

Sample Response (Success)

```
[
  {
    "deviceId": "ABC-123",
    "applicationKey": "myApp",
    "applicationType": "IOS",
    "additionalParameters": [
      {
        "key": "location",
        "value": "darmstadt"
      },
      {
        "key": "propertyType",
        "value": "house"
      }
    ],
    "created": "31.10.2015 12:04:21",
    "modified": "31.10.2015 13:59:04",
    "active": true ↵
  },
  {
    "deviceId": "QAY-692",
    "applicationKey": "myApp",
    "applicationType": "IOS",
    "additionalParameters": [
      {
        "key": "location",
        "value": "berlin"
      },
      {
        "key": "propertyType",
        "value": "flat"
      }
    ],
    "created": "31.10.2015 12:06:08",
    "modified": "31.10.2015 13:07:57",
    "active": false ↵
  }
]
```

- ↵ This device is still registered to receive notifications from the application.
- ↵ This device has unregistered from receiving the application's notifications.

5. Notification Interface

The notification interface provides services to send notifications to devices that have registered to receive these notifications. Notifications can be sent either as a push notification using the OS specific push notification system, or as a SMS/text message.

5.1. Diagnostic Headers

For diagnostic purposes both the Send Push Notification and the Send SMS interface accept diagnostic data from the caller and return a header containing the internal TPNS request ID to the caller.

Diagnostic data provided by the caller will be stored in the TPNS logs. The following diagnostic headers are currently supported:

Header Name	Sample Header Value	Description
X-TPNS-Application-Key	de.tpns.sampleApp	The application key of the calling application. A maximum of 200 characters will be accepted.
X-TPNS-Support-ID	7a6d6812-f95a-4da6-9e6c-181a7a117c7b	Arbitrary ID that allows identification of the sender (e.g. Box ID). A maximum of 200 characters will be accepted.

Responses from TPNS will return the internal request ID that is assigned by TPNS to each request. The internal request ID uniquely identifies every incoming request and links the request to the logs kept inside TPNS.

Header Name	Sample Header Value	Description
X-TPNS-Request-ID	8a52ee59-ff50-499b-9f62-7ae6a05418af	The unique TPNS request ID assigned to each incoming request. Please include this ID in all support requests.

Including this diagnostic data in any support request will provide the TPNS support team with an easy way to locate all logs related to the request.



Due to the high space requirement, logging data is kept for only a few days. The current retention period is 7 days.

5.2. Send Push Notification

The Send Push Notification service provides an API for sending of notifications by external systems to devices registered to receive their notifications. The service uses the operating system specific native push notification interfaces provided by the respective operating system manufacturers and supports the specific features of each notification system.

This service is capable of sending the same notification to devices with different operating systems while still supporting the specific features of each native notification system. The sending application may provide a list of devices to which to send the notification, or implicitly specify the devices by providing a list of filter criteria that are matched against the additional parameters sent with the device registration request (see [Device Registration Request Body](#)).

The service will queue all notifications internally and process them asynchronously. Thus the response to the sending application may be returned before the notification has actually been pushed to the devices.



External systems accept only notifications until certain payload size. Because of that TPNS also only accepts requests where title, message and custom fields have less than 3500 bytes. If those properties are bigger than that, error is returned.

5.2.1. Request

The send push notification request has the following format:

```
POST {SERVER_URL}/api/pushnotifications
```

5.2.2. Request Parameters

This service does not use any request parameters, however the application's password has to be provided in the HTTP-header:

Table 18. Send Push Notification Headers

Header Name	Header Value	R/O	Description
api.pnserver.password.access	string(255)	R	The password of the application specified in the <i>applicationKey</i> parameter in the request body.

5.2.3. Request Body

The request body must be a JSON object containing a **list of messages** described in the following tables.

Table 19. Send Push Notification Request Body

Property Name	Property Type	R/O	Description
applicationKey	string(255)	R	The ID of the sending application.
message	string	O	The message to be sent to the devices.
title	string	O	The title to be sent to the devices.
handlingLevel	integer	O	Priority in which the Notification will be processed. Valid values are 1-3. If no handlingPriority is given, the default value, of the application, is used.
highPriority	Boolean	O	Priority for FCM and HMS. True = high priority. False = normal priority. Default is normal priority.
applicationTypes	string[]	O	List of application types (see Supported Application Types) used when selecting the devices the message will be sent to.
deviceIds	string[]	O	List of devices that should receive the notification. The devices contained in the list may have different operating systems.
filters	filter[]	O	List of filters to be used to select the devices from the list of devices registered for the application. See Device Filters
additionalParams	object	O	The additionalParams is used to specify additional operating system specific values to be sent to the devices. For each supported operating system a specific parameter object is provided. See Additional Parameters . For an explanation of the OS specific parameters please refer to the documentation provided by the respective manufacturers.
dataPush	boolean	O	Flag to force data (AOS) or silent (IOS) pushes when set to true. By default, the value is set to false.

The *additionalParams* object holds a OS type specific parameter object as described in the following table:

Table 20. Additional Parameters

Property Name	Property Type	R/O	Description
iosParameters	object	O	Object to provide additional values used for iOS devices. See Additional Parameters for iOS Devices .

Property Name	Property Type	R/O	Description
aosParameters	object	O	Object to provide additional values used for Android devices. See Additional Parameters for Android Devices .
wosParameters	object	O	Object to provide additional values used for Windows devices. See Additional Parameters for Windows 10 Devices .
admParameters	object	O	Object to provide additional values used for Amazon devices. See Additional Parameters for Web Pushes .
webParameters	object	O	Object to provide additional values used for browser pushes. See Additional Parameters for Browser Pushes .
hmsParameters	object	O	Object to provide additional values used for Huawei Devices. See Additional Parameters for Huawei Devices .

The *iosParams* objects may contain additional parameters valid for sending of notifications to iOS devices only. For a detailed description please refer to the iOS specific documentation provided by Apple.

Table 21. Additional Parameters for iOS Devices

Property Name	Property Type	R/O	Description
customFields	customField[]	O	List of custom fields. See customField object .
badgeNumber	string	O	The number to display as the badge of the app icon.
soundFile	string	O	<p>The name of a sound file to play as an alert. TPNS will pass this value to APNS exactly as provided in the request. If an empty string is provided in the request, an empty string will be passed to APNS. If the <i>soundfile</i> property is omitted completely, TPNS will omit it in the call to APNS as well.</p> <p>A request where the <i>soundFile</i> property is omitted will lead to a push message without sound and without vibration, while a request with an empty <i>soundFile</i> property (<i>soundFile:""</i>) will result in a push message without sound, but with vibration.</p> <p>Specifying an unknown <i>soundFile</i> (not bundled with the app) will cause the device to play the default sound configured on the device.</p>
critical	boolean	O	The critical alert flag. If alert flag is true, then <i>soundFile</i> and <i>volume</i> are mandatory. It also means that critical interruption-level is set. Please mind that it can only work correctly, if app has required critical alerts entitlement.
volume	float	O	The volume for the critical alert's sound. Set this to a value between 0 (silent) and 1 (full volume). It is only used, when value of critical flag is true.
messageExpiryTime	string	O	Time in ms after which the message is considered as expired and not sent to the devices anymore.

Property Name	Property Type	R/O	Description
contentAvailable	string	O	Provide a value of "1" to send background notification and inform the application that new content is available. A background notification does not display an alert, play a sound or badge your app's icon.
mutableContent	string	O	Provide a value of "1" to notify the application that the notification should be modified by a notification service app extension. This allows the sender to enhance the notification with rich content (e.g. images, videos). It is the application's responsibility to correctly handle the rich content sent with the push.
category	string(64)	O	The category of this message. Up to 64 characters are accepted.
localizationKey	string	O	A key to an alert-message string for the current localization (set by the user's language preference). The alert-message string can contain placeholders to take the variables specified in the localizationArgs array.
localizationArgs	string	O	Array of string values to appear in place of the placeholders in the alert-message specified by localizationKey
collapseKey	string	O	An identifier (apns-collapse-id) you use to coalesce multiple notifications into a single notification for the user. The value of this key must not exceed 64 bytes.

The *aosParams* objects may contain additional parameters valid for sending of notifications to Android devices only. For a detailed description please refer to the aOS specific documentation provided by Google.

Table 22. Additional Parameters for Android Devices

Property Name	Property Type	R/O	Description
collapseKey	string	O	An identifier of a group of messages that can be collapsed, so that only the last message gets sent when delivery can be resumed. A maximum of 4 different collapse keys is allowed at any given time.
customFields	customField[]	O	List of custom fields. See customField object .
androidNotification	object	O	AndroidNotification Object. See AndroidNotification object .

The androidNotification object may contain additional parameters valid for sending of notifications to Android devices with Firebase Cloud Messaging support only. For a detailed description please refer to the Firebase Cloud Messaging Service specific documentation provided by Google Firebase.

Table 23. AndroidNotification object

Property Name	Property Type	R/O	Description
title	string	O	The notification's title.
body	string	O	The notification's body text. If present, it will override the message attribute.

Property Name	Property Type	R/O	Description
icon	string	O	Customized small icon on the left of a notification message. The icon file must be stored in the /res/raw directory of an app.
color	string	O	Custom notification bar button colors in the #RRGGBB format.
sound	string	O	Customized message notification ringtone, which is valid during channel creation. The ringtone file must be stored in the /res/raw directory of an app.
tag	string	O	Message tag. Messages that use the same message tag in the same app will be overwritten by the latest message.
click_action	string	O	The action associated with a user click on the notification. If specified, an activity with a matching intent filter is launched when a user clicks on the notification.
body_loc_key	string	O	ID in a string format of the localized message body.
body_loc_args	string[]	O	Variable parameter of the localized message body.
title_loc_key	string	O	ID in a string format of the localized message title.
title_loc_args	string[]	O	Variable parameter of the localized message title.
channel_id	string	O	Customized channel for displaying notification messages.

For additional details please see the [FCM documentation](#).

The *wosParams* objects may contain additional parameters valid for sending of notifications to Windows devices only. For a detailed description please refer to the Windows Notification Service (WNS) specific documentation provided by Microsoft.

Table 24. Additional Parameters for Windows 10 Devices

Property Name	Property Type	R/O	Description
type	string	O	Possible values: TILE, TOAST, BADGE, RAW. Note: RAW messages do not have a particular display style or predefined payload format and must be processed by the receiving app.
toastParams	object	O	Object to provide additional values for TOAST notifications. See Additional Parameters for Windows 10 TOAST Pushes . Ignored if type <> TOAST.
badgeParams	object	O	Object to provide additional values for BADGE notifications. See Additional Parameters for Windows 10 BADGE Pushes . Ignored if type <> BADGE.
rawParams	object	O	Object to provide additional values for RAW notifications. See Additional Parameters for Windows 10 RAW Pushes . Ignored if type <> RAW.
cache	boolean	O	Enables or disables notification caching (see WNS header X-WNS-Cache-Policy).

Property Name	Property Type	R/O	Description
timeToLive	integer	O	Integer value, expressed in seconds, that specifies the time to live (see WNS header X-WNS-TTL).
statusRequest	boolean	O	Enables or disables the request of device status and WNS connection status in the notification response (see WNS header X-WNS-RequestForStatus). The additional data provided by Microsoft is logged in TPNS and available on request.

Table 25. Additional Parameters for Windows 10 TOAST Pushes

Property Name	Property Type	R/O	Description
launch	string	O	A string that is passed to the application when it is activated by the toast. The format and contents of this string are defined by the app for its own use.
duration	string	O	The amount of time the toast should display. This attribute can have one of the following values: long, short
text2	int	O	
commands	string	O	Specifies that the toast notification is being used to indicate an incoming call or an alarm, with appropriate commands associated with each scenario. See Commands Object for Windows 10 TOAST Pushes

Table 26. Commands Object for Windows 10 TOAST Pushes

Property Name	Property Type	R/O	Description
scenario	string	O	The intended use of the notification. This attribute can have one of the following values: alarm, incomingCall
commands	list	O	A list of scenario-associated buttons shown in a toast. See Command for Windows 10 TOAST Pushes

Table 27. Command for Windows 10 TOAST Pushes

Property Name	Property Type	R/O	Description
id	string	O	A command from the system-defined command list. This attribute can have one of the following values: snooze, dismiss, video, voice, decline
arguments	string	O	An argument string that can be passed to the associated app to provide specifics about the action that it should execute in response to the user action.

Table 28. Additional Parameters for Windows 10 BADGE Pushes

Property Name	Property Type	R/O	Description
value	string	R	Specifies a badge's value or glyph. Valid values are either a numeric value or one of the predefined badge glyphs ("none", "activity", "alert", ..)
version	integer	O	The version of the badge XML schema this particular payload was developed for.

Table 29. Additional Parameters for Windows 10 RAW Pushes

Property Name	Property Type	R/O	Description
content	string	O	The body of the notification in app-defined format. The client receives the data as a null-terminated string (HSTRING) that only needs to be understood by the app.

The *admParams* objects may contain additional parameters valid for sending of notifications to Amazon devices only. For a detailed description please refer to the specific documentation provided by Amazon.

Table 30. Additional Parameters for Web Pushes

customFields	customField[]	O	List of custom fields. See customField object.
consolidationKey	string(64)	O	This is an arbitrary string used to indicate that multiple messages are logically the same and that ADM is allowed to drop previously enqueued messages in favor of this new one. Note that there are no guarantees that the previously enqueued messages will not be delivered. Your consolidation key may be no greater than 64 characters in length.
expiresAfter	int	O	The number of seconds that ADM should retain the message if the device is offline. After this time, the message may be discarded. Allowed values range from 60 (1 minute) to 2678400 (31 days), inclusive. The default value is 604800 (1 week).
md5	string	O	This is a base-64-encoded MD5 checksum of the data parameter. For details see Amazon Device Messaging documentation.

The *webParams* objects is used to provide additional parameters valid for sending browser push notifications only. Please note that browser pushes are not yet fully standardized and not every browser will support every feature yet. For a detailed description please refer to the resources available on the internet, e.g. introductions provided by [W3C](#), [Mozilla](#) or [Google](#).

Table 31. Additional Parameters for Browser Pushes

Property Name	Property Type	R/O	Description
title	string	M	The notification's title. This is the only mandatory attribute required to create a notification.
body	string	O	The notification's body text
dir	String	O	Direction of the text flow. Valid values are <i>auto</i> , <i>ltr</i> , and <i>rtl</i> .
lang	string	O	Specifies the language used in the in the notification. Either a valid BCP 47 language tag or the empty string. Examples: <i>en-US</i> , <i>de-DE</i>
image	string	O	URL of an image to be displayed in the notification.
icon	string	O	The URL of an image to be used as an icon by the notification.
badge	string	O	The URL of an image to represent the notification. Currently used in the Android Notification Bar only.

Property Name	Property Type	R/O	Description
vibrate	long[]	O	A vibration pattern applied when showing the notification. The values specify times in ms, numbers at even positions (0, 2, 4, ..) indication how long to vibrate, numbers at odd positions (1, 3, ..) indication how long to pause between vibrations.
silent	boolean	O	A value of <i>true</i> indicates that no sounds or vibrations should be made.
tag	string	O	Used as grouping key. Notifications visible to the user will automatically be replaced by new notifications with the same tag, instead of creating additional notifications.
renotify	boolean	O	A boolean value that indicates whether to suppress vibrations and audible alerts when reusing a tag value. The default is false.
requireInteraction	boolean	O	Indicates that on devices with sufficiently large screens, a notification should remain active until the user clicks or dismisses it. If this value is absent or false, the desktop version of Chrome will auto-minimize notifications after approximately twenty seconds. The default value is false.
actions	Object[]	O	An array of actions to display in the notification. Each element in the array may consist of an action, a title, and an associated icon. It is the responsibility of the application to respond to the click events. See action element for details.
customFields	customField[]	O	Arbitrary data to be associated with the notification. See customField object .

Table 32. Action element properties

Property Name	Property Type	R/O	Description
action	string	O	An arbitrary string identifying a user action to be displayed on the notification.
title	string	O	Action text to be shown to the user.
icon	string	O	The URL of an icon to display with the action title.

For a detailed explanation of these options please also refer to the documentation available on [MDN](#).

For a working example see [Sample Request for Browser Push](#).

The *hmsParams* objects may contain additional parameters valid for sending of notifications to Huawei devices only. For a detailed description please refer to the specific documentation provided by Huawei.

Table 33. Additional Parameters for Huawei Devices

Property Name	Property Type	R/O	Description
collapseKey	integer	O	An identifier of a group of messages that can be collapsed, so that only the last message gets sent when delivery can be resumed. 0: Only the latest offline message sent by each app to the user device is cached. -1: All offline messages are cached. 1-100: Offline message cache group ID. Offline messages are cached by group. Each group can cache only one offline message for each app.
customFields	customField[]	O	List of custom fields. See customField object . With customFields and without notification message is a data message.
androidNotification	object	O	AndroidNotification Object. See AndroidNotification object .

The androidNotification object may contain additional parameters valid for sending of notifications to Huawei devices with Huawei Mobile Services similarly as to Android devices. For a detailed description please refer to the HMS Push Kit specific documentation provided by Huawei.

Table 34. AndroidNotification object

Property Name	Property Type	R/O	Description
title	string	R	Android notification message title. Value is sent to hms as message.android.notification.title (title from Request Body is sent as message.notification.title). It means that value in AndroidNotification has to be used, when sending to android devices.
body	string	R	Android notification message body. Value is sent to hms as message.android.notification.body (message from Request Body is sent as message.notification.body). It means that value in AndroidNotification has to be used, when sending to android devices.
icon	string	O	Customized small icon on the left of a notification message. The icon file must be stored in the /res/raw directory of an app.
color	string	O	Custom notification bar button colors in the #RRGGBB format.
sound	string	O	Customized message notification ringtone, which is valid during channel creation. The ringtone file must be stored in the /res/raw directory of an app.
default_sound	boolean	O	Indicates whether to use the default ringtone
tag	string	O	Messages that use the same message tag in the same app will be overwritten by the latest message.
click_action	clickAction	R	Message tapping action. See clickAction object .
body_loc_key	string	O	ID in a string format of the localized message body.
body_loc_args	string[]	O	Variable parameter of the localized message body.
title_loc_key	string	O	ID in a string format of the localized message title.
title_loc_args	string[]	O	Variable parameter of the localized message title.

Property Name	Property Type	R/O	Description
channel_id	string	O	Customized channel for displaying notification messages. Customized channels are supported in the Android O version or later.

Table 35. *clickAction* object

Property Name	Property Type	R/O	Description
type	integer	R	Message tapping action type. The options are as follows: 1: custom tapping action. 2: tap to open a specified URL. 3: tap to start the app. 4: tap to access rich media information.
intent	string	O	When type is set to 1, you must set at least one of intent and action.
url	string	O	URL to be opened. This parameter is mandatory when type is set to 2.
rich_resource	string	O	URL of a rich media resource to be opened. This parameter is mandatory when type is set to 4.
action	string	O	Action corresponding to the activity of the page to be opened when the custom app page is opened through the action. When type is set to 1, you must set at least one of intent and action.

Table 36. *customField* object

Property Name	Property Type	R/O	Description
key	string(255)	R	The name of the custom field
value	string(255)	R	The value of the custom field

Sample Request

```
[
  {
    "applicationKey": "MYPUSHAPP",
    "message": "This message will be sent to all devices",
    "highPriority": true,
    "handlingLevel": 1,
    "applicationTypes": [],
    "deviceIds": [],
    "additionalParams": {
      "aosParameters": {},
      "iosParameters": {
        "customFields": [
          {
            "key": "par1",
            "value": "val1"
          },
          {
            "key": "par2",
            "value": "val2"
          }
        ]
      },
      "badgeNumber": "1",
      "soundFile": "marimba",
      "messageExpiryTime": "100",
      "contentAvailable": "1",
      "category": "WARN",
      "localizationKey": "ALERT_MESSAGE_KEY",
      "localizationArgs": ["someValue", "someOtherValue"]
    },
    "wosParameters": {
      "type": "TILE",
      "tileBackgroundImage": "someString",
      "tileCount": "2",
      "tileBackBackgroundImage": "someOtherString",
      "tileBackTitle": "someTitle",
      "tileBackContent": "stringAgain",
      "toastSubtitle": "subtitleString",
      "toastParameter": "toastParamString"
    }
  },
  {
    "filters": [
      {
        "key": "location",
        "value": "berlin"
      }
    ]
  }
]
```

To send a JSON object as a value in the customFields array, the object has to be converted to a string and all quotation marks need to be escaped by a backslash (\) character.

To add the object

```
{
  "included":{
    "type":"people",
    "id":"42",
    "attributes":{
      "name":"John",
      "age":80,
      "gender":"male"
    }
  }
}
```

as value, the following string representation has to be used:

```
"{\\"included\\":{\\"type\\":\\"people\\",\\"id\\":\\"42\\",\\"attributes\\":{\\"name\\":\\"John\\",\\"age\\":80,\\"gender\\":\\"male\\"}}}"
```

The request can then be used like so:

Sample Request with JSON object as value

```
[
  {
    "applicationKey":"MYPUSHAPP",
    "message":"This message will be sent to all devices",
    "highPriority":true,
    "handlingLevel":1,
    "applicationTypes":[ ],
    "deviceIds":[ ],
    "additionalParams":{
      "iosParameters":{
        "customFields":[
          {
            "key":"jsonValue",
            "value": "{\\"included\\":{\\"type\\":\\"people\\",\\"id\\":\\"42\\",\\"attributes\\":{\\"name\\":\\"John\\",\\"age\\":80,\\"gender\\":\\"male\\"}}}"
          }
        ]
      }
    }
  }
]
```

To send a message with rich content to an IOS device, include the *mutableContent* flag and provide the additional content in then *customFields* array:

Sample Request with rich content

```
[
  {
    "applicationKey": "MYPUSHAPP",
    "message": "This message will be sent to all devices",
    "highPriority": true,
    "handlingLevel": 1,
    "applicationTypes": [],
    "deviceIds": [],
    "additionalParams": {
      "iosParameters": {
        "mutableContent": "1",
        "customFields": [
          {
            "key": "image_url",
            "value":
              "https://cdn.pixabay.com/photo/2016/10/17/07/53/busan-night-scene-
              1747130_640.jpg"
          }
        ]
      }
    }
  }
]
```



Figure 1. Push Message with Rich Content

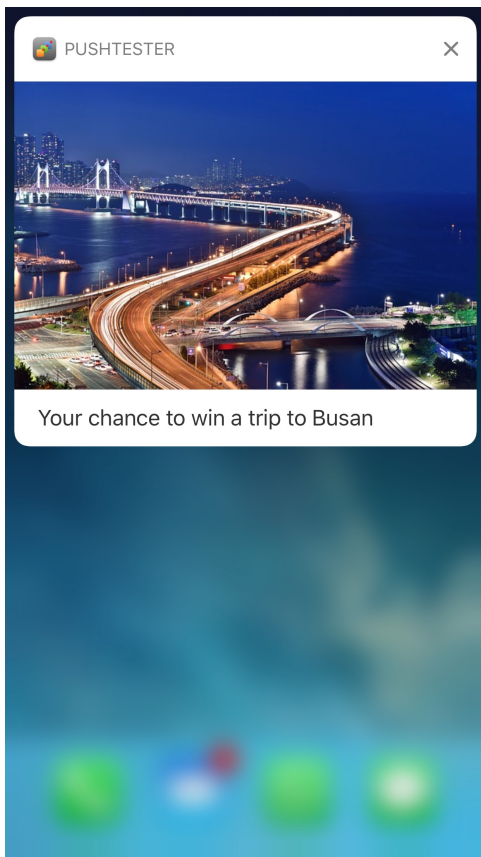


Figure 2. Rich Content From Push Message

The following example shows how to send a browser push with an image from an external URL and two action buttons. Properties specific to browser pushes have to be provided in the *webParameters* as shown below:

Sample Request for Browser Push

```
[
  {
    "applicationKey": "MYPUSHAPP",
    "applicationTypes": [ "WEB" ],
    "deviceIds": [],
    "filters": [],
    "handlingLevel": 1,
    "highPriority": true,
    "message": "We have exciting news for you.\nClick below to read on..",
    "additionalParams": {
      "webParameters": {
        "title": "Hot and exiting",
        "image": "https://images.all-free-
download.com/images/graphicthumb/small_chili_hot_picture_165462.jpg",
        "tag": "421963f2-2fa9-48f7-80df-72834b24d2b7",
        "actions": [
          { "action": "read", "title": "Sure, take me there" },
          { "action": "ignore", "title": "Hm, maybe later" }
        ]
      }
    }
  }
]
```

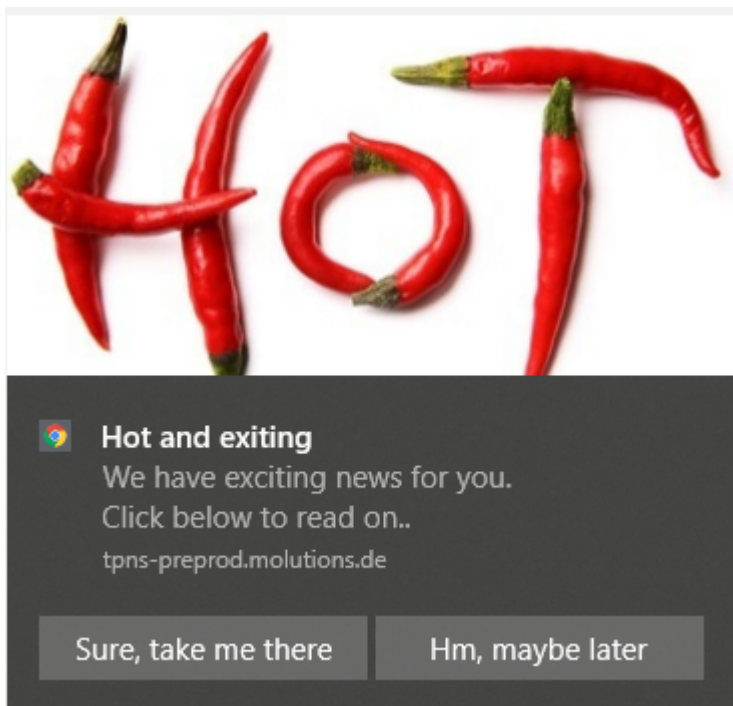


Figure 3. Browser Push Message



Not all browsers support all features for browser pushes. The result of the example push may look different on different browsers.

Upon completion, an HTTP status of 200 will be sent back to the calling application, together with a response object indicating the processing result. The calling application has to check the response object in order to detect an unsuccessful execution of service call.

Table 37. Send Push Notification Response

HTTP Status	Description
200	The operation completed.
403	Forbidden. This error code is returned when an invalid API key (see Send Push Notification Headers) is used, or when either the application or the customer owning the application is locked. The exact reason will be provided in the <i>errors</i> field contained in the response body.

5.2.4. Response Body

Table 38. Send Push Notification Response Body

Property Name	Property Type	R/O	Description
success	boolean	R	Indicates if the send SMS request is completed successfully ("true") or is failed ("false"). If the service call has failed, the <i>errors</i> element will contain a list of error messages.
message	string	R	A message providing a general error description (e.g. internal server error). If a list of specific errors is provided in the <i>errors</i> element, this field may be empty.
errors	array	R	List of errors that were detected during processing the send push notification request. Empty if the request completed successfully. See table errors Array



A successful service call will only inform the caller that the request to TPNS did not contain any errors (like unknown application types, unknown device IDs, missing data). It does not guarantee that the messages will be delivered to the devices.

Table 39. errors Array

Property Name	Property Type	R/O	Description
field	string	R	The name of the field that caused the error
message	string	R	A message describing the error
errorCode	string	R	An error code for the occurred error

Sample Response (Success)

```
{
  "success": true,
  "message": "All messages will be processed and sent",
  "errors": null
}
```

Sample Response (Failure)

```
{
  "success": false,
  "message": "Invalid notification request",
  "errors": [
    {
      "field": "deviceIds",
      "message": "unknown deviceId F7D322AD-016D-4F0C-8040-A118DA50469C",
      "errorCode": "UNKNOWN_DEVICE_ID"
    }
  ]
}
```

Sample Response (Failure because message is bigger than allowed)

```
{
  "success": false,
  "message": "Invalid notification request",
  "errors": [
    {
      "field": "messages",
      "message": "Message size exceeded",
      "errorCode": "INVALID_MESSAGE"
    }
  ]
}
```

5.3. Send SMS

This service allows sending of a text message to a single device.

Before an application can send SMS messages via TPNS, it needs to be configured as a SMS application using

the TPNS front-end. The 'Send SMS' requests for applications that are not configured correctly will be rejected.

5.3.1. Request

The send SMS request has the following format:

```
POST {SERVER_URL}/api/sendsms
```

5.3.2. Request Parameters

This service does not use any request parameters, however the application's password has to be provided in the HTTP-header:

Table 40. Send SMS Headers

Header Name	Header Value	R/O	Description
api.pnserver.password.access	string(255)	R	The password of the application specified in the <i>applicationKey</i> parameter in the request body. <i>If no API-Key is given the validation will be skipped.</i>

5.3.3. Request Body

The request body must contain a JSON object as described in the following table.

Table 41. Send SMS Request

Property Name	Property Type	R/O	Description
applicationKey	string(255)	R	The ID of the sending application
phoneNumber	string(9-16)	R	The destination address for the message. Only MSISDN as destination is allowed. The phoneNumber shall include the country code preceded by '+'. MSISDN length is min 9, max 16.
message	string(765)	R	Textual short message content. Messages over 70 characters may end up being sent as two or more messages. Maximum message length is 765 characters. Encoding UCS-2. The use of any character not specified in the Encoding may affect the delivery of the SMS.
senderAddress	string(9-16)	R	The address of the sender. Either MSISDN or a shortcode as sender address are allowed. For MSISDN the country code preceded by '+' has to be included. MSISDN length is min 9, max 16. Shortcode (if used) is passed 'as is' (not preceded with 'tel:'), e.g. '42424'. Shortcode length is min 1, max 8.
senderName	string	R	Name of the sender to appear on the user's terminal as the originator of the message. Supported characters: 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 (Spaces)'. The use of any other character may affect the delivery of the SMS (request gets rejected, originator altered or delivery failed).
senderID	String	O	An optional identifier. Used to identify which client (SmartHome-Box, etc.) sent the message.

Property Name	Property Type	R/O	Description
charset	String(GSM7/UCS2)	O	<p>The character set used to send the message. Supported values are GSM7 and UCS2.</p> <p>When GSM7 is used, messages up to 160 characters will be sent as a single SMS. Messages longer than 160 characters will be split into multiple SMS with 153 characters per SMS.</p> <p>Messages that require characters that are not supported by the 7 bit GSM7 character set may use the UCS2 16 bit character set instead. When UCS2 is used the maximum length of a single SMS is 70 characters. Messages longer than 70 characters will be split into multiple messages of 65 characters each.</p> <p>If no character set is specified, UCS2 will be used as the default value.</p>

Sample Request

```
{
  "applicationKey": "smsApp",
  "phoneNumber": "+1130247115342",
  "message": "This is a message",
  "senderAddress": "+420603603603",
  "senderName": "MYSMSAPP",
  "senderID": "1234556-1234565-12334432",
  "charset": "UCS2"
}
```

5.3.4. Response

Upon completion, an HTTP status of 200 will be sent back to the calling application, together with a response object indicating the processing result. The calling application has to check the response object to detect an unsuccessful execution of service call.

Table 42. Send SMS Response

HTTP Status	Description
200	The operation is completed.
403	This code is returned if the application has not been configured as an SMS application or the configuration is incomplete.

5.3.5. Response Body

Table 43. Send SMS Response Body

Property Name	Property Type	R/O	Description
success	boolean	R	Indicates if the send SMS request has been completed successfully ("true") or has failed ("false"). If the service call has failed, the <i>errors</i> element will contain a list of error messages.

Property Name	Property Type	R/O	Description
message	string	R	A message providing a general error description (e.g. internal server error). If a list of specific errors is provided in the <i>errors</i> element, this field may be empty.
errors	array	R	List of errors that were detected during processing the service request. Empty if the request completed successfully. See table errors Array .

Table 44. errors Array

Property Name	Property Type	R/O	Description
field	string	R	The name of the field that caused the error
message	string	R	A message describing the error
errorCode	string	R	An error code for the occurred error

Sample Response (Success)

```
{
  "success": true,
  "message": "All messages will be processed and sent",
  "errors": []
}
```

Sample Response (Failure)

```
{
  "success": false,
  "message": "Invalid notification request",
  "errors": [
    {
      "field": "phoneNumber",
      "message": "The phoneNumber may not be missing/empty",
      "errorCode": "INVALID_PHONENUMBER"
    },
    {
      "field": "message",
      "message": "The message may not be missing/empty",
      "errorCode": "INVALID_MESSAGE"
    }
  ]
}
```

Sample Response (Application not configured for SMS)

```
{
  "success": false,
  "message": null,
  "errors": [
    {
      "field": "applicationKey",
      "message": "No SMS provider configured for application smsApp",
      "errorCode": "SMS_PROVIDER_MISSING"
    }
  ]
}
```

6. Phone Number Validation Interface

The phone number validation interface provides a service to verify that a mobile phone number is a valid T-Mobile phone number. It will not provide any other information about this phone number.

6.1. Phone Number Validation

This service allows the verification of valid mobile phone numbers provided by T-Mobile.



Internally this service will call a further service provided by T-Mobile in order to hand over the request data and to perform the validation. A value of *"true"* in the *success* property of the response only indicates that the T-Mobile service could be called successfully. Whether the given phone number is a valid T-Mobile phone number, the result's *message* property has to be analyzed.

6.1.1. Request

The phone number validation request has the following format:

```
POST {SERVER_URL}/api/validatephonenumber
```

6.1.2. Request Parameters

This service does not use any request parameters, however the application's password has to be provided as a header:

Table 45. Validate Phone Number Headers

Header Name	Header Value	R/O	Description
api.pnserver.password.access	string(255)	R	The password of the application specified in the <i>applicationKey</i> parameter in the request body.

6.1.3. Request Body

The request body must contain a JSON object as described in the following table.

Table 46. Validate Phone Number Request Body

Property Name	Property Type	R/O	Description
applicationKey	string(255)	R	The ID of the sending application
phoneNumber	string(15)	R	The phone number to validate. The phone number has to be a valid MSISDN and consist of digits only. A preceding '+' is not required and will be ignored. The maximum MSISDN length is 15 digits.

Sample Request

```
{
  "applicationKey": "smsApp",
  "phoneNumber": "1130247115342"
}
```

6.1.4. Response

Upon completion, an HTTP status 200 will be sent back to the calling application, together with a response object indicating the processing result. The calling application has to check the response object in order to decide whether the phone number is a valid one.

Table 47. Validate Phone Number Response

HTTP Status	Description
200	The operation completed.

6.1.5. Response Body

Table 48. Validate Phone Number Response Body

Property Name	Property Type	R/O	Description
success	boolean	R	<p>Indicates if the validate phone number request has been completed successfully (<i>"true"</i>) or has failed (<i>"false"</i>).</p> <p>If the service completed the request successfully, the <i>message</i> property will contain one of the following messages:</p> <ul style="list-style-type: none"> - "Valid phone number" to indicate that the phone number is a T-Mobile phone number. - "Invalid phone number" to indicate that the phone number is not a valid T-Mobile phone number or unknown. <p>If the service call is failed, the <i>errors</i> element will contain a list of error messages.</p>
message	string	R	A message providing a general error description (e.g. internal server error). If a list of specific errors is provided in the <i>errors</i> element, this field may be empty.

Property Name	Property Type	R/O	Description
errors	array	R	List of errors detected during the service request processing. Empty (<i>null</i>) if the request is completed successfully. See table errors Array .
additionalInformation	object	O	This object contains additional information about the result of the validation request. It should be used for diagnostic purposes only. See table additional information .

Table 49. additional information

Property Name	Property Type	R/O	Description
providerId	string	O	The ID of the provider returned from CRISP. This field will be missing in case of an error. For the list of provider names please see the list of known providers .
faultCode	string	O	Error code returned from CRISP in case the call to CRISP was not successful or the validated phone number is not known in CRISP. This field will be missing if no error occurred.
errorCode	string	O	The description of the error code. This field will be missing if no error occurred.

Table 50. errors Array

Property Name	Property Type	R/O	Description
field	string	R	The name of the field that caused the error
message	string	R	A message describing the error
errorCode	string	R	An error code for the occurred error

Table 51. List Of Known Providers

Provider ID	Provider Name
1	T-Mobile
2	debitel
3	The Phonehouse
5	MobilCom/Cellway
6	Talkline
7	Telco
10	Alphatel
11	VictorVox
15	LiftaCom
17	Test-SP
21	klarmobil
22	callmobile
24	Simply/congstar Services GmbH
28	congstar

Provider ID	Provider Name
31	ensercom
35	Lebara
40	easyTel
42	Turkcell Europe GmbH

Sample Response (Success, T-Mobile Phone Number)

```
{
  "success":true,
  "message":"Valid phone number",
  "errors":null,
  "additionalInformation": {
    "providerId": "1"
  }
}
```

Sample Response (Failure, Not A T-Mobile Phone Number)

```
{
  "success": false,
  "message": "Invalid phone number",
  "errors": null,
  "additionalInformation": {
    "faultCode": "subscriberNotKnown",
    "faultText": "The subscriber is unknown to the TDG system."
  }
}
```

Sample Response (Failure)

```
{
  "success": false,
  "message": "Invalid phone number validation request",
  "errors": [
    {
      "field": "phoneNumber",
      "message": "The phoneNumber may not be missing/empty",
      "errorCode": "INVALID_PHONENUMBER"
    }
  ]
}
```