Description:  
Scope:

* SMS API
* Migrate the SMS API to the target ASE-design
* Migrate from SMS-provider Vonage (<https://developer.kpn.com/products/vonage-messages-api>) to provider KPN (<https://developer.kpn.com/products/kpn-sms-api>)
* FYI: Vonage used to be called 'Nexmo'
* Update the confluence-page: <https://confluence.kpn.org/spaces/SNIO/pages/300747186/SMS+API>
* Update the SMS API documentation
* Create test-cases
* BA-tests & approval will be done by our own team.

Acceptance Criteria:

* The SMS API is migrated to the target ASE-design
* The SMS API is connected to provider KPN (in stead of Vonage)
* The confluence-page has been updated.
* The SMS API documentation has been updated.
* Test-cases are available.

The following is listed on the confluence page:

[SMS API](https://confluence.kpn.org/spaces/SNIO/pages/300747186/SMS+API)

Information will be added

**API's:**

* SMS KPN Rest API
* SMS KPN SOAP API

Only ServiceNow Black is still using the SMS KPN SOAP API

Default Sender ID: KPNB2B

SMS Provider Vonage: <https://developer.kpn.com/products/vonage-sms-api>

**Example outbound JSON-message from SN Green to SMS KPN Rest API:**

{

"caseid" : "jjansen@ABC",

"mobilephones" : [ "+31612345678" ],

"operator" : "KPN",

"smsmessage" : "Your new temporary password for the ServiceNow with validation code 01234567 is: \*\*\*\*\*\*\*\*\*\* ",

"source" : "SN"

}

Vonage / **Messages**

**Elevate customer experience with rich social chats**

Easily send messages via Vonage Messages

Nexmo is now called Vonage, but there are still references to Nexmo in our URLs, code snippets and message templates.

Vonage's Messages API is a gateway to popular social chat apps such as WhatsApp Business, Facebook Messenger, Viber and SMS. It takes the complexity of hosting and scaling messaging and WhatsApp software off your hands. The API allows you to send notifications, customer service messages and other essential information your customers need on your customer's favorite messaging channel. For instance, leverage the WhatsApp for Business location feature to send them location information and enrich your messages with images, audio, video and files. All without having to worry about the privacy and security of your chats. This way you can focus on elevating your customer's experience.

Enrich your messages with location information, images, audio, video and files. For example, a customer waiting for a delivery can share their current location via WhatsApp, and have it delivered where they need it. If the package arrives damaged, the customer can simply send a photo.  
  
**SMS**

GDPR compliance: Yes  
SLA: Standard via the KPN Developer Portal

**Specifications**

API Type: REST  
Coverage: Available for telephone numbers globally  
Data center location: The Netherlands

**SMS Documentation**

Waarschuwing

Some countries might require additional information before being able to send SMS messages

**Introduction**

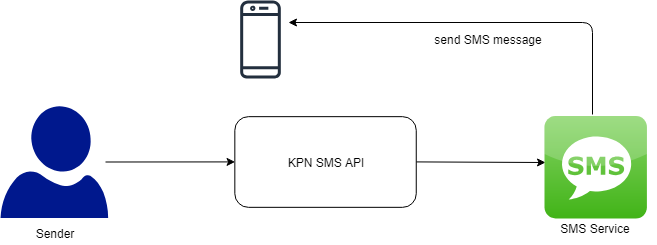
With the SMS API, anyone can build reliable SMS messaging services using our infrastructure. It is available as a free demo version too. Send some test SMS messages and experiment with the API before taking it into production. In your Developer Portal account you can apply to take the SMS API into production. Bulk SMS is now supported.

**API specification**

**Base URL**

https://api-prd.kpn.com/communication/kpn/sms

**Conceptual model**

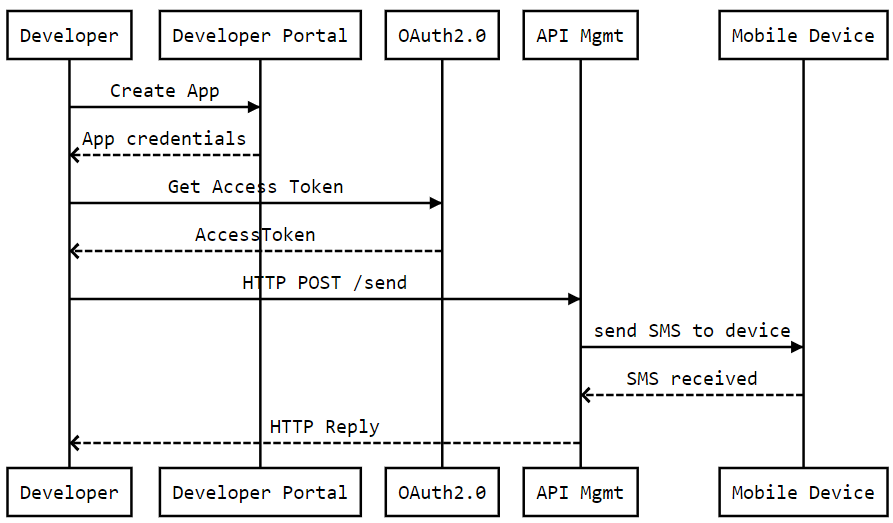


**Definitions**

**SMS**

An abbreviation of Short Message Service. It's a service on mobile phones to send and receive short messages.

**API workflow**



**Features and constraints**

**Features**

* Send a text message to another mobile phone number.
* Bulk messaging is now supported: multiple message can be sent to a single recipient, as well as a single message to multiple recipients. This combined features: multiple messages to multiple recipients.

**Constraints**

* No images can be sent with SMS.
* Messages longer than 160 characters will be sent as multiple SMS messages. Maximum message length is 1000 characters.
* With this API, messages can be sent globally to mobile phones. Also 097 range is allowed.
* With bulk messaging, the maximum number of messages multiplied by number of recipients in a single API call is 200.
* In sandbox mode, message length is capped to 160 chars. This limitation is of course lifted in the production version.

060xxxxxxx, 067xxxxxxx and 069xxxxxxx are not valid mobile numbers.

**Getting started**

Make sure you've read [What's in it for you](https://developer.kpn.com/page/whats-in-it-for-you) for more info on how to register and start testing APIs.

**Authentication**

The API follows the KPN Store API Authentication Standard to secure the API. It includes the use of OAuth 2.0 client\_id and client\_secret to receive an access token.

Go to the Authentication tab on top of this page to find out how to:

* Authenticate to an API using cURL.
* Authenticate to an API on Swaggerhub.
* Import Open API Specifications (OAS), also called Swagger files into Postman.

**How to...**

**Send a single SMS**

Send an SMS by calling the send endpoint of the SMS API in Swaggerhub or Postman.

**post:**{BASE\_URL}/send

Create your payload for the request using below snippet:

Request example

{

"sender": "KPN API",

"expirein": "5",

"messages": [

{

"mobile\_number": "06xxxxxxxx or +316xxxxxxxx",

"content": "Hi from KPN!"

}

]

}

The payload uses following parameters:

| **Parameter** | **Description** |
| --- | --- |
| sender | A text that should resemble the sender's origin. This string can have a maximum length of 11 characters. |
| mobile\_number | The mobile phone number of the addressee. Use the relevant country code at the start. For example, +31 or +44. |
| content | Put your message here. Long messages will be split into multiple SMS. |
| expirein | (Optional) The expirein field sets a time limit, in seconds, for how long the message can be delivered before it expires and is no longer sent. |

SwaggerHub:

1. Select POST /send.
2. Click Try it out.
3. Edit the body parameter by providing the payload snippet above. In the payload change the content, mobile\_number and sender to your own good. Make sure the content-type is set to application/json.
4. Click Execute.
5. Check the response code and message.

Postman:

1. Select (POST) /send.
2. In the Body section, set the type to raw and insert the payload snippet above. In the payload change the content, mobile\_number and sender to your own good. Make sure the content-type is set to application/json.
3. Click Send.
4. Check the response code and message.

Result example:

Response example

{

"document\_id": "b4e905d4-774c-4c83-8360-01427e17a33a",

"status": "OK"

}

**Sending bulk SMS's**

Sending bulk SMS's uses the same endpoint as a single SMS: the POST /send endpoint of the SMS API in Swaggerhub or Postman. Bulk SMS's are multidimensional, which means that you can:

* send a single SMS to multiple recipients,
* send multiple SMS's to a single recipient,
* a combination of these two options; send multiple SMS's to multiple recipients.

Example of a payload for the request with a single SMS being send to multiple recipients:

Request example

{

"sender": "KPN API",

"messages": [

{

"mobile\_number": "06xxxxxxxx,+316yyyyyyyy,06zzzzzzzz",

"content": "Hi from KPN!"

}

]

}

Example of a payload for the request with a multiple SMS's being send to multiple recipients:

Request example

{

"sender": "KPN API",

"messages": [

{

"mobile\_number": "06xxxxxxxx",

"content": "Hi from KPN!"

},

{

"mobile\_number": "+316xxxxxxxx",

"content": "Hello there."

},

{

"mobile\_number": "06xxxxxxxx,06yyyyyyyy,+316zzzzzzzz",

"content": "Till we meet again."

}

]

}

Please keep in mind that the maximum message length is 1000 characters. Maximum messages times recipients in one API call is capped to 200 messages.

You will be charged for every SMS you send to a recipient. In the example above where one of the messages is sent to 3 recipients, will be charged as 1 + 1 + 3\*1 = 5 SMS's.

This will also work in your sandbox. Feel free to test this, but be aware that you have a limited quota of 25 messages for testing purposes.

The payload uses following parameters:

| **Parameter** | **Description** |
| --- | --- |
| sender | A text that should resemble the sender's origin. This string can have a maximum length of 11 characters. |
| mobile\_number | The mobile phone number(s) of the addressee(s). Use the country code +31 at the start. |
| content | Put your message here. Long messages will be split into multiple SMS. |

SwaggerHub:

1. Select POST /send.
2. Click Try it out.
3. Edit the body parameter by providing the payload snippet above. In the payload change the content, mobile\_number and sender to your own good. Make sure the content-type is set to application/json.
4. Click Execute.
5. Check the response code and message.

Postman:

1. Select (POST) /send.
2. In the Body section, set the type to raw and insert the payload snippet above. In the payload change the content, mobile\_number and sender to your own good. Make sure the content-type is set to application/json.
3. Click Send.
4. Check the response code and message.

Result example:

Response example

{

"document\_id": "b4e905d4-774c-4c83-8360-01427e17a33a",

"status": "OK"

}

**Receive notification**

For each SMS sent, you can receive a notification. For this you'll need a webhook configured to receive these notifications. By sending the URL of this webhook along with the request, the notification will be delivered on this URL, where you can process this. This applies for single SMS and for bulk SMS. With bulk SMS you can set this up to:

* receive all notifications in the bulk request on a single URL,
* receive a notification for one or more recipients on a specific URL,
* receive a notification for one or more messages on a specific URL,
* or make a combination of above options; receive notifications on a specific URL for one or more messages, one or more recipients and still use a general URL for the remainder of the messages.

Receiving notifications is free of charge.

Here are a couple of examples of a payload for the request with SMS's being send with the setup of a webhook.

One webhook for one or all SMS

{

"sender": "KPN",

"webhook\_url": "https://hostname/path",

"messages": [{

"content": "Hi from KPN!",

"mobile\_number": "+316xxxxxxxx"

}],

}

Webhooks for each SMS

{

"sender": "KPN",

"messages": [{

"content": "Hi from KPN!",

"mobile\_number": "+316xxxxxxxx",

"webhook\_url": "https://hostname1/path1"

},{

"content": "Hi from KPN!",

"mobile\_number": "+316xxxxxxxx",

"webhook\_url": "https://hostname2/path2"

}],

}

A mix – one SMS with a dedicated webhook and others with a general webhook.

{

"sender": "KPN",

"webhook\_url": "https://hostname/path",

"messages": [{

"content": "Hi from KPN!",

"mobile\_number": "+316xxxxxxxx"

},{

"content": "Another Hi from KPN!",

"mobile\_number": "+316xxxxxxxx",

"webhook\_url": "https://hostname2/path2"

},{

"content": "Yet another hi from KPN!",

"mobile\_number": "+316xxxxxxxx"

}],

}

Dedicated webhook(s) have a priority over the general one.

On the webhook you are going to receive a json message. The structure is as follows:

Notification received example

{

"Fields": {

"Status": {

"StatusDateTime": "2021-12-02T13:25:34Z",

"StatusValue": "1",

"StatusCode": "Delivered"

},

"Sender": "KPN",

"WebhookURL": "https://hostname/path",

"Message": "Hi There",

"RecipientPhonenumber": "+316xxxxxxxx",

"CustomerId": "email@domain.com"

},

"DLRID": "9F7742AD353FSGS2345SDCDSV45C02",

"Service": "Sms",

"MessageType": "SmsDeliveryReport",

"MessageID": "4C0FWFWE86EE492ABCD34534122hJKHKHFDW"

}

Error Notification received example:

{

"Fields": {

"Status": {

"StatusDateTime": "2021-12-02T13:25:34Z",

"StatusValue": "4",

"StatusCode": "Failed",

"ErrorValue": "2",

},

"Sender": "KPN",

"WebhookURL": "https://hostname/path",

"Message": "Hi There",

"RecipientPhonenumber": "+316xxxxxxxx",

"CustomerId": "email@domain.com"

},

"DLRID": "9F7742AD353FSGS2345SDCDSV45C02",

"Service": "Sms",

"MessageType": "SmsDeliveryReport",

"MessageID": "4C0FWFWE86EE492ABCD34534122hJKHKHFDW"

}

The possible values of StatusCodes are:

* 1 - Delivered
* 2 - Queued
* 3 - Accepted
* 4 - Failed
* 5 - Rejected
* 6 - Expired

The possible values of ErrorValues are:

* 1 - Unknown
* 2 - Absent Subscriber - Temporary
* 3 - Absent Subscriber - Permanent
* 4 - Call Barred by User
* 5 - Portability Error
* 6 - Anti-Spam Rejection
* 7 - Handset Busy
* 8 - Network Error
* 9 - Illegal Number
* 10 - Illegal Message
* 11 - Unroutable
* 12 - Destination Unreachable
* 13 - Subscriber Age Restriction
* 14 - Number Blocked by Carrier
* 15 - Prepaid Insufficient Funds
* 16 - Gateway Quota Exceeded
* 17 - Message Blocked by Provider
* 50 - Entity Filter
* 51 - Header Filter
* 52 - Content Filter
* 53 - Consent Filter
* 54 - Regulation Error
* 99 - General Error

Delivered: The "Delivered" status confirms that the message has successfully reached the recipient's mobile device. It means that the message has passed through all the necessary network channels and has been received by the recipient's phone. This status is a strong indicator that the intended recipient has the opportunity to read the message.

Queued: The "Queued" status indicates that the message is queued for delivery to the intended recipient. The message is still with the message processor.

Accepted: The "Accepted" status indicates that the message processor ran all checks succesfully, correct number format, valid content etc. The message will continue to be processed.

Failed: The "Failed" status indicates that the message was not successfully sent to the recipient. This could be due to a variety of reasons such as incorrect phone numbers, network issues, or the recipient's phone being turned off or out of service. When a message fails, it means that it never reached the intended recipient's device.

Rejected: The "Rejected" status indicates an error for example with the message, recipient phonenumber or missing registrations in the recipients country.

Expired: The expired status indicates that the message failed to be delivered within the SMS lifespan. And won’t be retried.

**Delivery Receipt / Notification (DLR) Retry Policy**

We try to send the DLR up to four times if it fails:

* **1st attempt:** Sent immediately when the status is available.
* **2nd attempt:** 60 seconds after the first attempt fails.
* **3rd attempt**: 120 seconds after the second attempt fails.
* **4th (final) attempt:** 240 seconds after the third attempt fails.

After four failed attempts, we stop retrying.

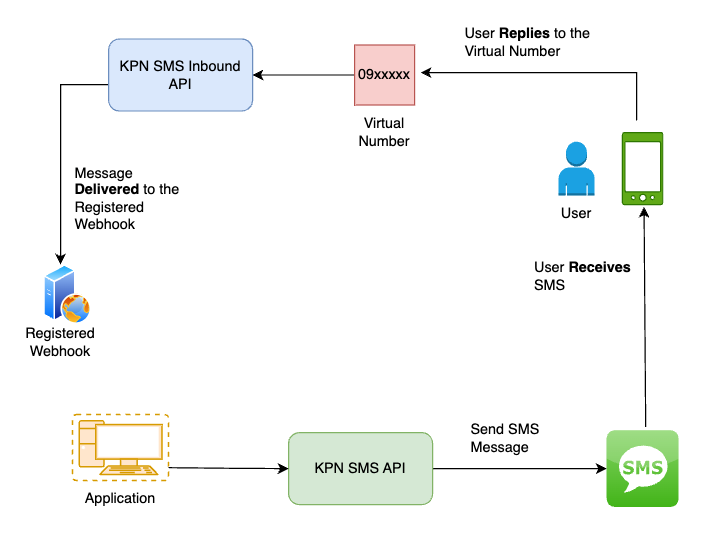
The **StatusDateTime** field in the notification shows the time when the delivery status was generated. It is always in epoch format.

**Inbound SMS**

What is Inbound SMS?

Inbound SMS refers to text messages sent to your virtual number. These messages can be used for customer queries, feedback, or receiving confirmation codes, making it a versatile tool for enhancing communication with your audience.

You can receive inbound SMS messages directly to your virtual number and a registered webhook. You will need a virtual number and an active webhook. You can request a virtual number and the webhook configuration by filling out the [support form](https://developer.kpn.com/support).



KPN / **SMS**

**Fast and reliable two-way text messaging**

Inform end users with SMS notifications such as confirmations, updates and alerts

Integrate fast and reliable SMS messaging into your application with the SMS API from KPN. Send updates, confirmations, reminders and other notifications to your customers and employees directly from your application. For instance, you could send proactive service message, send out a promotion, alert an employee and securely log in with 2-factor authentication.

Reach users globally in any language. You can easily integrate it into your own software and use SMS to send timely notifications and to boost conversion. You pay per use only, so no investment is needed.

How it works

Through the SMS REST API you are connected to KPN’s Short Message Peer-to-Peer (SMPP) protocol. In this way you don't need to develop and support a protocol yourself. Send SMSes by sending calls to the API. In the API call you can insert your content and set the right settings for your purpose. Your SMS messages are always delivered quickly and reliably via KPN's network. Want to see how it works? Check out our documentation page.