Security & Encryption

TLS Requirements

It is recommended that the URL domain is compatible for both testing and production to ensure that during the testing stage, notification configuration meets PayNet requirements.

INFO

Our APIs only support **TLS 1.2**

Signature Generation

INFO

You can also find our message signature SDK or sample apps in **resources** section.

The following steps describe how a message signature is generated. The following list of message body parameters and values will be used for this example:

Parameter Name	Parameter Value
payerbanknum	100033061
payerbankname	AGROBANK
billerbanknum	100002144

Parameter Name	Parameter Value
billerbankname	STANDARD CHART. BANK
accounttype	1
billercode	1123
billercodename	Maxis
nbpsref	3598D772
channel	3
debittimestamp	2016-05-09T08:07:39.0000000+00:00
repeatmsg	N
rrn	1230
rrn2	
currencycode	MYR
amount	4550
extdata	null

Step 1: Retrieve the timestamp

The **message timestamp** for the operation will need to be retrieved in UTC time, e.g. **2016-05-09T08:07:05.2199272Z** from message header.

Step 2: Rearrange the Message Body Parameters

All the parameters of the **message body** are sorted alphabetically using their parameter names. Using the parameter list above as an example, the result would be as follows:

Parameter Name	Parameter Value
accounttype	1
amount	4550
billerbankname	STANDARD CHART. BANK
billerbanknum	100002144
billercode	1123
billercodename	Maxis
channel	3
currencycode	MYR
debittimestamp	2016-05-09T08:07:39.0000000+00:00
extdata	null
nbpsref	3598D772

Parameter Name	Parameter Value
payerbankname	AGROBANK
payerbanknum	100033061
repeatmsg	N
rrn	1230
rrn2	

Step 3: Concatenate the Parameter Values

Each of the message body parameter values are joined together to form one string. Any empty parameters can be left out from their place in the order.

```
1 + 4550 + STANDARD CHART. BANK + 100002144 + 1123 + Maxis + 3 + MYR + 201
05-09T08:07:39.0000000+00:00 + 3598D772 + AGROBANK + 100033061 + N + 1230
```

This will result in the following string being generated:

```
14550STANDARD CHART. BANK1000021441123Maxis3MYR2016-05-
09T08:07:39.0000000+00:003598D772AGROBANK100033061N1230
```

Step 4: Insert the timestamp in front of the concatenated parameter values

The timestamp obtained from Step 1 is attached to the concatenated string from Step 3 as a prefix with the following format: yyyyMMddHHmmss

The definition of the format is as follows:

Format	
yyyy	4 digit Year
MM	2 digit Month, therefore single digit months are padded with a zero in front, e.g. February is 02
dd	2 digit Day, single digit days are padded with a zero
HH	2 digit Hour in 24 hour format, zero padded
mm	2 digit Minute, zero padded
ss	2 digit Seconds, zero padded

Give the example above, the time stamp of **2016-05-09T08:07:05.2199272Z** will become: `20160509080705`

This string is placed as a prefix in front of the parameter values obtained in Step 3, which results in the following:

```
2016050908070514550STANDARD CHART. BANK1000021441123Maxis3MYR2016-05-09T08:07:39.0000000+00:003598D772AGROBANK100033061N1230
```

Step 5: Insert the Pass Key provided to the Biller at the end of the string

Each **Biller** is provided a secret **Pass Key** for use with BNS. For this example, we will use the pass key = **"JQDSHALPKX"**, and this value will be appended to the end of the string obtained from Step 4 as a suffix. This will result in the following string:

```
20160509080705145 50STANDARD CHART. BANK1000021441123Maxis3MYR2016-05-09T08:07:39.0000000+00:003598D772AGROBANK100033061N1230JQDSHALPKX
```

Step 6: Calculate SHA256 of the string and encode the results in Base64

The SHA256 hash of the string obtained from Step 5 encoded in Base64 is:
`EFddam1Y/dvhBh2Nvytw2XvZMQHB3sONuWeEMzP9CuA=`

This is **message signature** for the message. It is either inserted in message header when generating a message, or validated against message signature contained in a message response.

Example Request Message

Using the fictional example above, the request message would be as follows:

```
{
  "header": {
    "sig": "EFddam1Y/dvhBh2Nvytw2XvZMQHB3sONuWeEMzP9CuA=",
    "timestamp": "2016-05-09T08:07:05.2199272Z"
    },
  "body": {
```

```
    "payerbanknum": "100033061",
    "payerbankname": "AGROBANK",
    "billerbanknum": "100002144",
    "billerbankname": "STANDARD CHART. BANK",
    "accounttype": "1",
    "billercode": "1123",
    "billercodename": "Maxis",
    "nbpsref": "3598D772",
    "channel": "3",
    "debittimestamp": "2016-05-09T08:07:39.0000000+00:00",
    "repeatmsg": "N",
    "rrn": "1230",
    "rrn2": "",
    "currencycode": "MYR",
    "amount": 4550,
    "extdata": null
    }
}
```


Sample Code

Java

```
package com.paynet.signer.ui.signer_jompay_ui;

import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
```

```
import java.util.Base64;

public class JomPaySignature {

  private final static String SHA256 = "SHA-256";

  public static String genSignature(String clearText) throws NoSuchAlgorithmExce
      return new String(

Base64.getEncoder().encode(MessageDigest.getInstance(SHA256).digest(clearText.ge
  }
}
```