BNI eCollection Technical

Specification v3.0.3

Document Sign-Off

Version 3.0.3		
Author	Candra	14 July 2016
Reviewer		
Approver		

Document Change Control

Version	Implementation Date	Authors	Summary of changes
1.0	23 June 2015	Nura Aditya	
1.1	24 June 2015	Nura Aditya	Revision the security data.
1.2	25 June 2015	Nura Aditya	API for bank client request.
1.3	22 September 2015	Nura Aditya	Fix format virtual account number, update flow process transaction, domain for access.
1.4	25 September 2015	Nura Aditya	Information for Client Portal System.
1.5	26 October 2015	Nura Aditya	Add information payment messaging for biller.
1.6	28 October 2015	Nura Aditya	Update flow architecture and process payment notification to biller.
1.7	14 January 2016	Candra	Added field `description` to the API; transaction amount can now be updated when the transaction has not been paid.
1.8	23 February 2016	Candra	Added field `billing_type` to the API; Changed `trx_amount` maximum length to 14.
1.8.1	16 Mei 2016	Candra	Renew outdated library links.
1.9.0	14 July 2016	Candra	All `datetime` parameters can now accept ISO8601 format; Added new parameter to `inquirybilling` response; Added new parameter to payment notification system; Added section 3.5 Parameter Information.

1.9.1	14 September 2017	Zola	Updated section 3.1; Added 2 (two) types of billing/invoice, minimum payment (m) and open minimum payment (n); Updated section 3.5, changed notes for parameter type, customer_name, customer_phone, va_status; Updated section 4, added status code 013, 107, 200, 201; Updated **Appendix C**; Updated **Appendix D**; Added library link for Node.js.
1.9.2	19 December 2017	Zola	Added 1 (one) types of billing/invoice, open maximum payment (x); Added library link for Golang.
3.0.0	27 April 2018	Zola	
3.0.1	14 Agustus 2019	Shella	Update 2.1. Virtual Account Number; Update 3.6 Parameter Information
3.0.2	09 September 2019	Shella	Update 3.1 Create Invoice/Billing add prefix on Request Update 3.2 Inquiry Invoice/Billing add prefix on Request Update 3.3 Update Transaction add prefix on Request
3.0.3	08 Oktober 2019	Shella	Update Encryption Library on Appendix and Parameter Information on billing_type Add Prefix on Parameter Information

Table of Contents

1. Introduction

Biller client is a merchant who integrated with the BNI eCollection system

BNI eCollection system to provide these services:

- Create invoice/billing (API)

- Inquiry invoice/billing details (API)

- Update invoice/billing details (API)

- Payment notification (callback)

2. Biller Client Integration

2.1 Biller Client Detail

Biller client should be able to provide technical data for the integration purpose, which consist:

- Company name
- Public IP address to be used to integrate with BNI eCollection
- A callback URL to receive payment notification from BNI eCollection
- Default billing lifetime, number of hours a billing can stay active

2.2 Virtual Account Number

The virtual account number has 16 digits long and consists of 3 components, below are the details:

Prefix VA	Client ID	Free Digit

- Prefix VA   : number that given by BNI
- Client ID   : number id client that is given by BNI along with the Secret Key
- Free Digit

2.3 Data Security System

For security purpose, all messages that sent from and to BNI eCollection will be encrypted with a library (provided, see **Appendix E**). The encryption algorithm needs a secret key which will be provided by BNI eCollection. The secret key is a regular 32 hexadecimal characters.

Sample of a secret key: **ea0c88921fb033387e66ef7d1e82ab83**

The encryption also uses time to validate the request. Biller client should be able to make sure the time on the machine which making a request is set properly, though there is a 5 (five) minutes tolerance.

3. Transaction Flow

Biller will interact with BNI eCollection web API using **JSON POST** on a single URL:

Development URL: **https://apibeta.bni-ecollection.com/**

Production URL: **https://api.bni-ecollection.com/**

These are the basic rules to integrate with BNI eCollection web API system:

- Using HTTP POST

- Header Content-Type should be application/json

- Request data should be encrypted with BNI eCollection encryption method

- Response data should be decrypted with BNI eCollection decryption method

Example request of creating an invoice/billing

Raw data:

```
{
    "client_id" : "001",
    "trx_amount" : "100000",
    "customer_name" : "Mr. X",
    "customer_email" : "xxx@email.com",
    "customer_phone" : "08123123123",
    "virtual_account" : "8001000000000001",
    "trx_id" : "1230000001",
    "datetime_expired" : "2015-07-01 16:00:00",
    "description" : "Payment of transaction ABC",
    "type" : "createBilling"
}
```

Encrypt the raw data before making a request, result of the encryption will look like this:

```
GEhHGEwbHh0WE0QNA0ZPTU1VCkVPejghNxFFGgQ9NVxVDnBBAVddWFYDUTQRFBMTGRc3E
wgDeFR5BE8KMANACTs3PgJ2XltdUk57BT9IUURRVDcgCQ0OYFRGA0lLfkBLUwIzDTVLXl
xWUQN3U0JTTFdVewggOEYfRhNIGhRERBoWOD0CClFaXldCA3FBR0ZSXlUJCSA3TxdEE0U
YE0FCGRNFQhBDGgoVBVUICz9MSAUiCkYYGkVGGEQRRhkEPjRcXQV2A00KTFtHQwp3I0xP
UFFVfQgSOHpICUYJUk92ck1bBnpSeUwKJAQTR0MVERMaFhdGBxdLURdEHEUYBQ4
```

Note: this is not the real result from encrypting the above JSON, because the result will vary over time

Send the encrypted data along with the provided Client ID:

```
{
    "client_id":"001",
    "prefix":"8",
    "data":"GEhHGEwbHh0WE0QNA0ZPTU1VCkVPejghNxFFGgQ9NVxVDnBBAVddWFYDU
            TQRFBMTGRc3EwgDeFR5BE8KMANACTs3PgJ2X1tdUk57BT9IUURRVDcgCQ
            0OYFRGA01LfkBLUwIzDTVLXlxWUQN3U0JTTFdVewggOEYfRhNIGhRERBo
            WOD0CClFaXldCA3FBR0ZSXlUJCSA3TxdEE0UYE0FCGRNFQhBDGgoVBVUI
            Cz9MSAUiCkYYGkVGGEQRRhkEPjRcXQV2A00KTFtHQwp3I0xPUFFVfQgSO
            HpICUYJUk92ck1bBnpSeUwKJAQTR0MVERMaFhdGBxdLURdEHEUYBQ4"
}
```

Example response of successful request:

```
{
    "status":"000",
    "data":"GEZHFEMcHh0WE0QNA1ZXRVxcCQggOEYXRQNBC0ZyB0kFUAwCClFaXld
            CA3FBR0ZSXlUJCSA3TxdEE0UYE0FCGRNFQhBDGgoVBVUICz9MSAUiCk
            YYGkVGGEQRRhkEDw8"
}
```

Decrypt the response data:

```
{
    "virtual_account" : "8001000000000001",
    "trx_id" : "1230000001"
}
```

Example response of failed request (will be sent unencrypted):

```
{
    "status" : "001",
    "message" : "Incomplete/invalid Parameter(s)."
}
```

3.1 Create Invoice/Billing

This section provides the information for Biller client to create an invoice/billing.

Accepted parameters to create an invoice/billing (for more information about each parameter, please refer to **section 3.6**).

Parameter	Mandatory	Notes
type	**Yes**	Set to createbilling or createbillingsms
client_id	**Yes**	
trx_id	**Yes**	Invoice/billing ID
trx_amount	**Yes**	
billing_type	**Yes**	See **Appendix A**
customer_name	**Yes**	
customer_email	No	
customer_phone	No	
virtual_account	No	
datetime_expired	No	
description	No	

Parameters of response from a successful request.

Parameter	Notes
trx_id	Invoice/billing ID
virtual_account	VA number will be generated by BNI eCollection system if it has not been provided on request parameters.

Note: Invoice/billing that has been created **cannot** be deleted, though it can be deactivated to prevent end-user from paying the associated VA number.

If parameter type is set to createbillingsms, BNI eCollection will also send an SMS to customer to pay the invoice/billing. SMS will be sent only if the provided number is registered with BNI SMS Banking.

Sample of `createbilling` request and response

Request:

```
{
    "type" : "createbilling",
    "client_id" : "001",
    "trx_id" : "1230000001",
    "trx_amount" : "100000",
    "billing_type" : "c",
    "customer_name" : "Mr. X",
    "customer_email" : "xxx@email.com",
    "customer_phone" : "08123123123",
    "virtual_account" : "8001000000000001",
    "datetime_expired" : "2016-03-01T16:00:00+07:00",
    "description" : "Payment of Trx 123000001"
}
--------------------------------------------------------------------
{
    "client_id":"001",
    "prefix":"8",
    "data":"GUtJFEMcHh0WE0QNA0ZPTU1VCkVPejghNxFFGgQ9NVxVDnBBAVddWFYD
            UTQRFBMTGRc3Ewh4DFoIUQJNVXAAS1B6NBo1NloXAzk4PwJGWVZcVwJL
            WXR7VXVKAgocNApgXFV2TnRRVRdFUQM0DQVGWVtbBVNLCHVXfVADTgRL
            NRgbR0MTRRobGxQUOT4CWkxVXVx2U0V2ekoDVwNcBUs0IRNFQxBDGRgZ
            ExFGQxATFQUUCg1YX3R_TDYbOBkURUIYFEVBEUQKFQtWWwZ3Ax0FR1pM
            d1pLWH9TAUoDUAQ9NUxECnZUfVVNSUdZB3tSSUcFIwlHFxdKRBdLD0UZ
            A0JIIxNFTBBDC2U"
}
```

Success response:

```
{
    "status":"000",
     "data":"FUpBFUMcHh0WE0QNA1ZXRVxcCQggOEYXRQNBC0ZyB0kFUAwCCl
            FaXldCA3FBR0ZSXlUJCSA3TxdEE0UYE0FCGRNFQhBDGgoVBVUI
            Cz9MSAUiCkYYGkVGGEQRRhkEDw8"
}
--------------------------------------------------------------------
{
    "status" : "000",
    "data" : {
        "virtual_account" : "8001000000000001",
        "trx_id" : "1230000001"
    }
}
```

3.2 Inquiry Invoice/Billing

This section provides the information for Biller client to retrieve information of the existing invoice/billing.

Accepted parameters to inquiry an invoice/billing (for more information about each parameter, please refer to **section 3.6**).

Parameter	Mandatory	Notes
type	**Yes**	Set to `inquirybilling`
client_id	**Yes**	
trx_id	**Yes**	Invoice/billing ID. This parameter will be used to identify which invoice/billing to be presented.

Parameters of response from a successful request.

Parameter	Notes
client_id	
trx_id	
trx_amount	
virtual_account	
customer_name	
customer_email	
customer_phone	
datetime_created	Deprecated. Please use `datetime_created_iso8601` instead.
datetime_created_iso8601	
datetime_expired	Deprecated. Please use `datetime_expired_iso8601` instead.
datetime_expired_iso8601	
datetime_last_updated	Deprecated. Please use `datetime_last_updated_iso8601` instead.
datetime_last_updated_iso8601	
description	
va_status	
payment_amount	
payment_ntb	
Billing_type	
datetime_payment	Deprecated. Please use `datetime_payment_iso8601` instead.
datetime_payment_iso8601	

Sample of `inquirybilling` request and response

Request:

```
{
    "type" : "inquirybilling",
    "client_id" : "001",
    "trx_id" : "1230000001",
}
-----------------------------------------------------------------------
{
    "client_id":"001",
    "prefix":"8",
    "data" : "GUpCFUMcHh0WE0QNA0ZPTU1VCkVPejghNxFFGgQ9NVxVDnBJeAoiDB
            MTSkIQFBMTGRg3EwgJEFd5BE8KTH8DXkwHCyJ8VVRSUUg4EA"
}
```

Success response:

```
{
    "status":"000",
    "data": "GkdDFUMcHh0WE0QNA1ZXRVxcCQggOEYXRQNBC0ZyB0kFUAwCCFpgSUt
            FOUwCFRUWGRdFFxZFSAlABAtRVQUHSk90c0N2WF1XVwNQNRgTFBQYGE
            UWF0VGGEQRRhkEPjRLWQgFUABNW0hQQwN3Ax0FMVoVNj4IQjhXdloCT
            lAFclZXeDMaNhgZHBUVTEgXHBwFFQl5SFp6C1ABR3RYRAp_TlEJNBo1
            GxgaGA5GSQ0VFwMaG08YGk9GITYNOFxUCnFJUQQGTwcKIwsTEkZCERM
            FEApXd19TewRbdEICWFd_BwodOEIQRBgYGgReFA"
}
-----------------------------------------------------------------------
{
    "status" : "000",
    "data" : {
        "client_id" : "001",
        "trx_id" : "1230000001",
        "trx_amount" : "100000",
        "virtual_account" : "8001000000000001",
        "customer_name" : "Mr. X",
        "customer_phone" : "08123123123",
        "customer_email" : "xxx@email.com",
        "datetime_created" : "2016-02-01 16:00:00",
        "datetime_expired" : "2016-03-01 16:00:00",
        "datetime_last_updated" : "2016-02-10 16:00:00",
        "datetime_payment" : "2016-02-23 23:23:09",
        "payment_ntb" : "023589",
        "payment_amount" : "100000",
        "va_status" : "2",
        "billing_type" : "c",
        "description" : "Payment of Trx 123000001",
        "datetime_created_iso8601" : "2016-02-01T16:00:00+07:00",
        "datetime_expired_iso8601" : "2016-03-01T16:00:00+07:00",
        "datetime_last_updated_iso8601" : "2016-02-10T16:00:00+07:00",
        "datetime_payment_iso8601" : "2015-06-23T23:23:09+07:00"
    }
```

```
}
```

3.3 Update Transaction

This section provides the information for Biller client to update details of the existing invoice/billing.

Accepted parameters to update an invoice/billing (for more information about each parameter, please refer to **section 3.6**).

Parameter	Mandatory	Notes
type	**Yes**	Set to updatebilling
client_id	**Yes**	
trx_id	**Yes**	Invoice/billing ID. This parameter will be used to identify which invoice/billing to be updated.
trx_amount	**Yes**	See **Appendix C**.
customer_name	**Yes**	
customer_email	No	
customer_phone	No	
datetime_expired	No	
description	No	

Parameters of response from a successful request.

Parameter	Notes
trx_id	Invoice/billing ID
virtual_account	VA number of the updated invoice/billing.

Note:

- In order to maintain invoice/billing detail, biller client must provide every value that are set before when requesting an update to BNI eCollection. If, for example, biller client doesn't provide customer_email in update request, then its previous value will be replaced with an empty string.
- Parameter billing_type and virtual_account **cannot** be updated.

Sample of updatebilling **request and response**

Request:

```
{
    "client_id" : "001",
    "trx_id" : "1230000001",
    "trx_amount" : "100000",
    "customer_name" : "Mr. X",
    "customer_email" : "xxx@email.com",
    "customer_phone" : "08123123123",
    "datetime_expired" : "2016-03-30T23:00:00+07:00",
    "description" : "Payment of Trx 123000001",
    "type" : "updateBilling"
}
--------------------------------------------------------------------------
{
    "client_id":"001",
    "prefix":"8",
    "data":"GUdEFUMcHh0WE0QNA0ZPTU1VCkVPejghNxFFGgQ9NVxVDnBJeAoiDB
            MTSkIQFBMTGRg3EwgJCV9zQwJXWH8GCx03QxBDGRgZBQ04dlVWWFJV
            TQdFVXYDTTYbODVUQDJABkEzRAhbXVhPRwhxRlBETVQJUAheDg4nek
            52Uk4_dldQOD0Cd11bXlFOfAQ_VEtSV0w3IQhFTxhGFUYaFkJEHAVB
            NER0XU1dTE57ckVbVExaTXkIITdIGEUWQxgYP0UYBEdEG0MYIxkSBE
            I0VVxTSQohOFtWendbeiN-VU56AU8FEw"
}
```

Success response:

```
{
    "status":"000",
    "data":"GkxEFUMcHh0WE0QNA1ZXRVxcCQggOEYXRQNBC0ZyB0kFUAwCClFaXl
            dCA3FBR0ZSXlUJCSA3TxdEE0UYE0FCGRNFQhBDGgoVBVUICz9MSAUi
            CkYYGkVGGEQRRhkEDw8"
}
--------------------------------------------------------------------------
{
    "status" : "000",
    "data" : {
        "virtual_account" : "8001000000000001",
        "trx_id" : "1230000001"
    }
}
```

3.4 Transaction Payment Notification (BNI eCollection system to Biller Client POST JSON)

Parameters of payment notification (for more information about each parameter, please refer to **section 3.6**).

Parameter	Notes
virtual_account	
customer_name	
trx_id	Invoice/billing ID
trx_amount	
payment_amount	
cumulative_payment_amount	
payment_ntb	
datetime_payment	
datetime_payment_iso8601	

Notes:

- **Callback URL:** provided by Biller client,

- If payment notification failed, our system will send another request to callback URL until succeed or at maximum 5 times retry,

- A notification is considered as success if and only if response from the callback URL indicated status = 000 with JSON format (see example on the next page),

- Payment simulation for development purpose: http://dev.bni-ecollection.com/.

Example of payment notification from BNI eCollection:

```
{
    "trx_id" : "1230000001",
    "virtual_account" : "8001000000000001",
    "customer_name" : "Mr. X",
    "trx_amount" : "100000",
    "payment_amount" : "100000",
    "cumulative_payment_amount" : "100000",
    "payment_ntb" : "233171",
    "datetime_payment" : "2016-03-01 14:00:00",
    "datetime_payment_iso8601" : "2016-03-01T14:00:00+07:00"
}
-------------------------------------------------------------------------
{
    "client_id":"001",
    "data":"FUdFFUMcHh0WE0QNA1dVXEdQegggOEcZSBFFGRJBQxkFQjNWfVpcX0N
            NdnNDR1JYV1s3IQhNRxdFEkUYE0FCGRNFQhBECxQLRlYJB09QSVVHVn
            ZTTDdQCmFTRAg6ND4KWAcJQHRVWF5QVjhMAxQTFBgXRggSOAZIDk56V
            1ZwdFVSC39UNiIKGxIRR0IQBg8FWUgOVEsDC0YCVncKHTNJGhVIQxJG
            CxQLR0IKeFRMUUhHWHZfVHoEXDYbOBoSQ0cVFEs-EkkIGh0cEkZMERM
            FYQ"
}
```

Response should be like this (otherwise it is considered as a failed request):

```
{
    "status" : "000"
}
```

3.5 Parameter Information

Parameters of payment notification (for more information about each parameter, please refer to **section 3.5**).

Parameter	Data Type	Notes
type	Accepted value: 　- createbilling 　- createbillingsms 　- updatebilling 　- inquirybilling	This field indicate your type of action, it is required for each API request. If this parameter is omitted, error code **005** will be thrown. If the provided value other than the accepted value, error code **004** will be thrown.
client_id	Char(2) or char(3) or char(5)	Client ID is given by BNI along with the Secret Key.
trx_id	varchar(30)	Invoice/billing ID. Accepts any **printable** characters. This parameter will be the identifier of each invoice/billing; thus, it must be unique. Invoice/billing of the same ID **cannot** be created more than once even if it has been expired, otherwise error code **105** will be thrown.
trx_amount	decimal(14,0)	Invoice/billing amount. Accepts only integer value without decimal and/or thousand separators. See **Appendix B** for more information.
billing_type	Accepted value: 　- o 　- c 　- i 　- m 　- n 　- x	See **Appendix A,** if merchant did not input billing type and amout : 0, will automatically open payment, if amount > 0 will be close payment

Parameter	Data Type	Notes
customer_name	varchar(255)	Accepts any **printable** characters. This parameter will be used as a VA name, which will be displayed when customer wants to pay the invoice/billing. If parameter type is set to createbillingsms, this parameter can only accept maximum 14 characters.
customer_email	varchar(255)	Accepts any **printable** characters with a valid email address format. This parameter will be used if biller client wants to send an invoice and/or payment email notification to customer.
customer_phone	varchar(30)	Accepts any **printable** characters. This parameter provide the customer's phone information. If parameter type is set to createbillingsms, this parameter should be started with **628**. Example: **628**1234567890
virtual_account	char(16)	VA number that will be used to make a payment. It must follow the format specified on **Section 2.2**.

Parameter	Data Type	Notes
datetime_expired	datetime Format: <date>T<time><timezone>	This parameter will determine VA's expiry date & time. It must follow the format specified on ISO 8601. Example: 2015-12-31T23:59:00+07:00 This parameter will use other date/time format if you do the inquirybilling request: <date> <time> Example: 2015-12-31 23:59:00 With this format, date & time are presented in UTC+7.
description	varchar(100)	Accepts any **printable** characters. This parameter provide some additional information about the invoice/billing, and will be displayed on the invoice email notification.
datetime_created	datetime Format: <date> <time>	This parameter presents only if you do the inquirybilling, indicating when the invoice/billing was created. Example: 2015-12-31 23:59:00 With this format, date & time are presented in UTC+7.

Parameter	Data Type	Notes
datetime_last_updated	datetime Format: <date> <time>	This parameter presents only if you do the inquirybilling, indicating when the invoice/billing was updated. Example: 2015-12-31 23:59:00 With this format, date & time are presented in UTC+7. This parameter is nullable.
datetime_payment	datetime Format: <date> <time>	This parameter presents if you do the inquirybilling and on payment notification. Indicating when the invoice/billing was paid. Example: 2015-12-31 23:59:00 With this format, date & time are presented in UTC+7. It will be null when the invoice/billing has not been paid.
datetime_created_iso8601	datetime Format: <date>T<time><timezone>	This parameter is the ISO 8601 version of datetime_created parameter. Example: 2015-12-31T23:59:00+07:00

Parameter	Data Type	Notes

`datetime_last_updated_iso8601`	`datetime`
	
	Format:
	`<date>T<time><timezone>`
		This parameter is the ISO 8601 version of `datetime_last_updated` parameter.
		
		Example:
		2015-12-31T23:59:00+07:00

`datetime_payment_iso8601`	`datetime`
	
	Format:
	`<date>T<time><timezone>`
		This parameter is the ISO 8601 version of `datetime_payment` parameter.
		
		Example:
		2015-12-31T23:59:00+07:00

`datetime_expired_iso8601`	`datetime`
	
	Format:
	`<date>T<time><timezone>`
		This parameter presents only if you do the `inquirybilling`, and is the ISO 8601 version of `datetime_expired` parameter.
		
		Example:
		2015-12-31T23:59:00+07:00

`payment_ntb`	`char(6)`	Journal number, also known as reference number. This parameter presents if you do the `inquirybilling` and on payment notification. It's a sequence of 6 (six) digit random number. Example: 058305 It will be `null` when the invoice/billing has not been paid.

Parameter	Data Type	Notes
payment_amount	decimal(14,0)	Paid amount. This parameter provide two different information from each of these scenarios: - in a response of inquirybilling request, it provides the **total** amount paid by end-user for the corresponding invoice/billing ID, - in payment notification system, this parameter indicates the **current** amount paid by the end-user.
cumulative_payment_amount	decimal(14,0)	This parameter presents only on payment notification, it provides the **total** amount paid by end-user for the corresponding invoice/billing ID.
va_status	integer	This parameter presents if you do the inquirybilling. It provide the status of the invoice/billing. 1 => invoice/billing is active 2 => invoice/billing is inactive
account_number	varchar(15)	Only for RDL Notification, This parameter presents account number of lender
accounting_flag	char(1)	Only for RDL Notification, This parameter presents C (credit) or D (debit)
Prefix	char(2) char(3) char(5)	This parameter must included in request encrypt

Note: printable characters are a set of all visible characters and spaces (having an ASCII hex between \x20 and \x7E).

4. Status Code Message

STATUS	MESSAGE
000	Success
001	Incomplete/invalid Parameter(s).
002	IP address not allowed or wrong Client ID.
004	Service not found.
005	Service not defined.
006	Invalid VA Number.
008	Technical Failure.
009	Unexpected Error.
010	Request Timeout.
011	Billing type does not match billing amount.
012	Invalid expiry date/time.
013	IDR currency cannot have billing amount with decimal fraction.
101	Billing not found.
102	VA Number is in use.
103	Billing has been expired.
105	Duplicate Billing ID.
107	Amount can not be changed.
200	Failed to send SMS Payment.
201	SMS Payment can only be used with Fixed Payment.
997	System is temporarily offline.
998	"Content-Type" header not defined as it should be.
999	Internal Error.

Appendix

A. There are 6 (six) types of invoice/billing that can be created:

- Open payment: invoice/billing can be paid <u>multiple times</u> as long as it is still active.

    `billing_type` code: **o**

- Fixed payment: invoice/billing should be paid with <u>exactly the same</u> amount as requested.

    `billing_type` code: **c**

- Installment/partial payment: invoice/billing can be paid <u>multiple times as long as paid amount is less than the requested amount</u> and still active.

    `billing_type code:` **i**

- Minimum payment: invoice/billing can be paid with <u>greater than or equal to</u> the requested amount.

    `billing_type` code: **m**

- Open minimum payment: invoice/billing can be paid <u>greater than or equal to</u> the requested amount <u>multiple times</u> as long as it is still active.

    `billing_type code:` **n**

- Open maximum payment: invoice/billing can be paid <u>less than or equal to</u> the requested amount <u>multiple times</u> as long as it is still active.

    `billing_type code:` **x**

B. Biller client should set `trx_amount` parameter to **0** when creating an open payment invoice/billing, otherwise error code **011** will be thrown.

C. Updating invoice/billing can only be done when it is never been paid before **or** the new amount is greater than or equal to paid amount, otherwise error code **107** will be thrown.

D. `cumulative_payment_amount` will differ from `payment_amount` when `billing_type` is set to **o** (open payment), **i** (partial payment), **n** (open minimum payment), or **x** (open maximum payment) due to the fact that customer can pay more than once.

E.   Encryption library:

    a.   ASP.NET:
        https://drive.google.com/drive/folders/1JvdFCmwICqLr2EObtIp58X-L9nIRnT2l

    b.   Java:
        https://drive.google.com/drive/folders/1bkCTAD8d7OpeA3tl214Sn8nKD1o6p1YP

    c.   PHP:
        https://drive.google.com/drive/folders/1U7Ot5cLfGG-nE7WMc6KrDn3FFZbfBlq_

    d.   Python:
        https://drive.google.com/drive/folders/1xREJ-tUSj2Wx9_WSRnz78CFmjnsd9nUV

    e.   Node.js:
        https://drive.google.com/drive/folders/1VcryGNOTvL73p-Pfks4nYNB66OcyoxCo

    f.   GO Lang:
        https://drive.google.com/drive/folders/10IvyqAD8QsjJEvDTGzEas6MD97PAWNKb