RESTful payment API

I. Method types

Method	Description	Sample URL of the request
pay_init	Check for due payment	https://example.com/pay/init
pay_confirm	Payment Notification	https://example.com/pay/confirm

II. Parameter types

Parameter	Data type	Description	Additional information	
IDN	varchar(64)n	Customer ID	Provided by the Merchant	
MERCHANTID	varchar(8)n	Merchant ID	Provided by the Operator	
INVOICE	varchar(64)ans	Invoice No	In the case of a separate payment of due payments	
INVOICES	varchar(490)ans	Database containing Invoice numbers	In the case of a separate payment of due payments	
SHORTDESC	varchar(40)Ans	Short description of the due payment	One row, coded in UTF-8	
LONGDESC	varchar(4000)Ans	Detailed description of the due payment	More than one row, coded in UTF-8	*
AMOUNT	integer	Amount in stotinki	Output check parameter	
VALIDTO	varchar(8)n	Date of the due payment	YYYYMMDD	
STATUS	varchar(3)n	Status or result of operation	Result or error code	
TID	varchar(26)n	Transaction ID	DATE(14)n + STAN(6)n + AID(6)n	*
DATE	varchar(14)n	Date of processing of the request	YYYYMMDDhhmmss	
TOTAL	integer	Amount in stotinki	Input payment parameter	
TYPE	varchar(7)a	CHECK, BILLING, PARTIAL, DEPOSIT	Depending on the request type	*
CHECKSUM	HEX	Input data, sorted alphabetically	IDNxxxx\nMERCHANTIDxxxx\n	*

LONGDESC Detailed description of the due payment. It is desirable to submit relevant information about the customer - period of the provided service, service type, address, name, other information. Must be submitted on one row, where the carrying over to the next row is coded with \n every 110 symbols.
It may contains the following substitutions:

\t	eight intervals
\$	eight dashes
\n	newline

INVOICE The invoices are specific term for the protocol, they are not related to real legal data. They are used for distinguishing accumulated more than one or separate due payments of one and the same customer. Each customers' due payment has a unique invoice.

TID Unique identifier for each transaction. Upon a problem with a certain transaction the Operator could submit more than one iteration until he receives correct response by the Merchant. The iteration has always one and the same TID.

- DATE Date and time, with accuracy to a second. The time can be from 00:00:00 to 23:59:59
- STAN Service information of the Operator - should not be processed
- AID Source of payment. There are two types - 70002x and 7001xx are cash payment by EasyPay cash desks. All other sources may be treated as ePay.bg electronic channels

CHECKSUM hmac_sha1_hex (request_data , SECRET), where request_data is concatenated with NEWLINE (\n) rows, containing a parameter and its relevant value - merged (IDN1234\nMERCHANTID0000334\nTOTAL100\n). The data is sorted in ascending order. SECRET is provided by the Operator.

TYPE	The field will contain one of the following values, according to the request type:
• CHECK	Only checking of the due payment. No payment will be conducted after the check.
• BILLING	Checking of the due payment, may be followed by payment.
• PARTIAL	Upon notification for partial payment
• DEPOSIT	Upon pre-payment of a service

III. STATUS codes

STATUS	Description	Method from which could be send back
00	OK	pay_init and pay_confirm
13	Invalid amount (for deposit)	pay_init
14	Invalid subscription number (IDN)	pay_init
62	No pending payments	pay_init
80	Temporary cannot be executed	pay_init
93	Invalid checksum (CHECKSUM)	pay_init and pay_confirm
94	Iteration of already sent notification	pay_confirm
96	General error	pay_init and pay_confirm

! Upon response, different than 00, all parameters except STATUS will be ignored. Only the meaning of the status predefined by the Operator will be visualised to the customer. The information in the fields as LONGDESC, SHORTDESC or AMOUNT will not be taken under consideration **!**

96 General error. Usually when the Operator's system hasn't received response by the Merchant in the scheduled period of 30 sec - the Operator interprets the lack of response as 96. Also, when there is invalid data in the output data fields, or missing compulsory fields in general, it is interpreted as 96. The Operator will iterate the notification until receiving the correct response by the merchant.

94 Has the meaning of 00. If the Merchant's system has accepted the payment request and has sent the status 00, but in Operator's system hasn't received the response, then as a rule the sender have to continue to resend the message until receiving correct response. When upon resending the same message, the Merchant can successfully submit code 94, which has the meaning of 00, then the request is successfully processed and the iterations may be stopped.

80 Temporary cannot be executed - The merchant temporary cannot process payments. Usually this code is sent when the merchant has temporary stopped processing payments because of updating of the amounts of the due payments by his customers or other specific reason.

I.I Checking for due payment

pay_init It is used for checking the current due payments of a certain customer. If the request does not contain field TID, no payment will be executed afterwards – this is only a check for presence of due payments. If the Merchant has developed payment of separate invoices, he has to return parameter INVOICES with the relevant data, when there are more than one due payment. This parameter is not submitted in the request when the Merchant offer payments of total amount only (does not support separate invoices payment).

! When upon request for receiving a payment, the merchant has returned response STATUS : 00 and amount higher than 0 in the output data, this a sign for the Operator that execution of the payment may be done and it will be processed. The merchant should be able to process the followed payment notification, type pay_confirm. If the payment couldn't be processed the merchant has to respond STATUS : 96 or other relevant status for the case, when responding on the pay_init method **!**

Input data

GET parameters	M/O	
IDN	M	Customer ID
MERCHANTID	M	Merchant ID
CHECKSUM	M	HMAC-SHA-1 HEX of the input data, sorted by key
TYPE	M	May contain CHECK or BILLING
TID	O	Is not present in requests with parameter TYPE=CHECK

Output data

JSON object	M/O	
STATUS	M	Upon response different than 00, all parameters except STATUS will be ignored
IDN	M	Customer ID
AMOUNT	M	Amount in stotinki. Upon payments in invoices, the total amount is submitted
VALIDTO	M	Date to which the due payment is current.
SHORTDESC	M	One row is submitted. Short information about the customer – name, address, other information
LONGDESC	M	One row is submitted. NEWLINE is coded with \n every 110 symbols
INVOICES	O	If the merchant do not support payments in separate invoices, this filed is not submitted.

Each separate invoice is submitted in a batch of objects INVOICES having the parameters described herein after. The invoice is formed by customer's subscription number, concatenated with dot and the invoice number (IDN.INVOICE) for the specific due payment.

Output data in the package INVOICES

JSON object	M/O	
IDN	M	Subscription number and invoice or the specific due payment, concatenated with dot (IDN.INVOICE)
AMOUNT	M	Amount of the specific invoice
VALIDTO	M	Date to which is current the specific invoice
SHORTDESC	M	Short description of the specific invoice
LONGDESC	M	Detailed description of the specific invoice

I.II Payment notification

pay_confirm It is used to notify the Merchant for executed payment by a customer. This notification will be received only after correct response of a request for extracting due payments type pay_init, where the field TID exist.

Payment total due payment If the merchant hasn't developed payments of separate invoices and submits only the total amount of the due payment, he will receive input data without field INVOICES.

Payment of invoices Upon payment of all invoices, the Merchant will receive notification without parameter INVOICES. The parameter TOTAL will contains the lump sum of all invoices returned by the Merchant in pay_init.

Upon payment of part of the submitted invoices, the Operator will send parameter INVOICES. Its value is formed by the customers subscription number, concatenated with dot and invoice number of the specific due payment (IDN.INVOICE). When there is more than one invoice, the values are separated with comma (IDN.INVOICE,IDN.INVOICE,IDN.INVOICE)

Partial payment In this case the Merchant will receive value TYPE=PARTIAL and the amount selected by the customer. It is possible this value to be lower than the value submitted with pay_init. This notification doesn't contain field INVOICES.

! The payment notification cannot be declined. The merchant will receive this message in specified time intervals, until he responses with STATUS : 00 or STATUS : 94 !

The merchant should be able to process several notification messages. i.e. one notification message may be received several consecutive times or if the merchant delays to send response of the first message (more than 30 sec.), he may receive second message while executing the first one. In any case only one payment message should be processed and for the rest STATUS 00 or 94 must be responded.

Input data

GET parameters	M/O	
IDN	M	Customer ID
MERCHANTID	M	Merchant ID
TID	M	Transaction ID
DATE	M	Date of execution of the request
TOTAL	M	Amount in stotinki
TYPE	M	May contain Billing or Partial
INVOICES	O	IDN.INVOICE – when there is more than one invoice the values are listed, separated by comma
CHECKSUM	M	HMAC-SHA-1 HEX of the input data, sorted by key

Output data

JSON object	
STATUS	All parameters except STATUS and its relevant predefined value, are ignored.

! When a problem occurs during confirmation of a transaction, the Operator repeats it automatically until receiving a correct response by the Merchant. The identifier by which the Merchant may understand that this is a repetition, not new payment, is TID – this parameter will always be one and the same for each repetition of a particular transaction **!**

IV.I Check deposit

pay_init Used for checking whether for a particular customer may be executing a deposit transaction. The request contains also an amount which the merchant may validate or not. The amount in the request may be for predefined nominees or randomly chosen by the customer.

! If in the response of the request for deposit validation, the merchant has responded STATUS : 00, this is a sign for the Operator that the payment may start and will be processed. The merchant should be able to process the next notification for payment, type pay_confirm.
If the merchant is unable to process a payment, he has to respond STATUS : 96 or other relevant status for that case, when responding on the pay_init method **!**

Inout data

GET parameters	M/O	
IDN	M	Customer ID
MERCHANTID	M	Merchant ID
CHECKSUM	M	HMAC-SHA-1 HEX of the input data, sorted by key
TYPE	M	DEPOSIT
TID	M	Transaction ID
TOTAL	M	Amount in stotinki

Output data

JSON object	M/O	
STATUS	M	If the value of the STATUS is not 00, all other fields will be ignored
SHORTDESC	M	Short description. Name, e-mail or other relevant client identifier
LONGDESC	M	One row is submitted. NEWLINE is coded with \n every 110 symbols

IV.II Notification for deposit

pay_confirm Used to notify the merchant for a processed deposit payment.

! The payment notification cannot be declined. The merchant will receive this message in specified time intervals, until he responses with STATUS : 00 or STATUS : 94 !

Input data

GET parameters	M/O	
IDN	M	Customer ID
MERCHANTID	M	Merchant ID
CHECKSUM	M	HMAC-SHA-1 HEX of the input data, sorted by key
TYPE	M	DEPOSIT
TID	M	Transaction ID
TOTAL	M	Amount in stotinki

Output data

JSON object	
STATUS	All parameters except STATUS and its relevant predefined value, are ignored.

V. Examples

All requests from the examples will be coded with SECRET: 3EA1ABD845C3D684

CHECKSUM Example for the value of request_data, sorted by ascending order before being coded. NEWLINE (\n) is present on the last row as well.

```
IDN12345
MERCHANTID0000334
TID20170317121650591535700020
```

Checking for due payment

- **Example for input data - HTTP GET (check for due payments only, no payment will occur)**

https://example.com/pay/init?
IDN=12345&CHECKSUM=702de02734d25c719c6ccc87526478e851f6271d&MERCHANTID=0000334&TYPE=CHECK

- **Example for input data - HTTP GET (payment may occur)**

https://example.com/init/?
IDN=12345&CHECKSUM=2736e17a183ed4b6923f7e0395b6c0523fdf0404&TID=20170317121650591535700020&MERCHANTID=00
00334&TYPE=BILLING

- **Example for output data - Json object with total amount**

```
{
  "STATUS" : "00",
  "IDN" : "12345",
  "SHORTDESC" : "John Doe, Internet service",
  "LONGDESC" : "Client info:\nClient number: 12345\nClient name: John Doe",
  "AMOUNT" : "16600",
  "VALIDTO" : "20170317",
 }
```

- **Example for output data - Json object with two invoices**

```
{
  "STATUS" : "00",
  "IDN" : "12345",
  "SHORTDESC" : "John Doe, Internet service",
  "LONGDESC" : "Client info:\nClient number: 12345\nClient name: John Doe\n Obligation period 01.03.2017 - 30.04.2017",
  "AMOUNT" : "16600",
  "VALIDTO" : "20170317",
  "INVOICES" : [
    {
      "IDN" : "12345.001",
      "SHORTDESC" : "John Doe, Internet service",
      "AMOUNT" : "7800",
      "LONGDESC" : "Buisness internet - 100 mbps 78 lv.\t\t\t| 31.03.2017 23:59:59 | 78.00   |\nClient name: John Doe\n+$$--+\n",
      "VALIDTO" : "20170331"
    },
    {
      "IDN" : "12345.002",
      "SHORTDESC" : "John Doe, Internet service",
      "AMOUNT" : "8800",
      "LONGDESC" : "Buisness internet - 100 mbps 88 lv.\t\t\t| 30.04.2017 23:59:59 | 88.00   |\nClient name: John Doe\n",
      "VALIDTO" : "20170430"
    }
  ]
}
```

Payment notification

- **Example for input data for total amount of the due payment**

https://example.com/confirm/?
DATE=20170316181226&TYPE=BILLING&MERCHANTID=0000334&IDN=12345&CHECKSUM=823383f09ab489fe172762703f8c047ce4428530&TOTAL=16600&TID=20170317121650591535700020

- **Example for input data for payment of one invoice**

https://example.com/confirm/?DATE=20170316181226&TYPE=BILLING&MERCHANTID=0000334&IDN=12345&TOTAL=7800&CHECKSUM=06c5786385a673bfcc25a10a6d59722769bca25f&TID=20170317121650591535700020&INVOICES=12345.001

- **Example for input data for partial payment**

https://example.com/confirm/?
DATE=20170316181226&TYPE=PARTIAL&MERCHANTID=0000334&IDN=12345&CHECKSUM=70514b288b2167b5bcf6324eaddc1a8179cebd57&TOTAL=100&TID=20170317121650591535700020

Check deposit

- **Example for input data for check deposit**

https://example.com/init/
IDN=12345&MERCHANTID=0000334&CHECKSUM=123c13322543764d4af33d87a4a8dd0965777ed6&TYPE=DEPOSIT&TID=20170317121650591535700020&TOTAL=2000

- **Example for output data for check deposit**

```
{
   "STATUS" : "00",
  "SHORTDESC" : "Client name: John Doe",
   "LONGDESC" : "1 Month prepaid subscription\nClient name: John Doe",
 }
```

Pay deposit

- **Example for input data for notification of pay deposit**

https://example.com/confirm/
IDN=12345&MERCHANTID=0000334&CHECKSUM=728094da1e3609abe5514d21604918e7b4877ca4&TYPE=DEPOSIT&TID=20170317121850591535700020&TOTAL=2000

- **Example for output data for pay deposit**

```
{
   "STATUS" : "00"
  }
```