

# MTS WebService Interface

---

## FLEX & Agnostic API

for using unattached EMV terminal  
with any environment  
or operating system

**ARKOM Computer Operated Systems LTD**

SMART CREDIT CARDS PROCESSING SYSTEMS

## MTS Webservice Interface

### FLEX & Agnostic API

The Web Service interface is intended for use with ARKOM's family of MTS clearing services.

It presents a new approach to full EMV implementation with just one line of code and no relevance to the operating system, language or environment.

Full EMV features connecting to un-plugged Terminal (no cables whatsoever) to any development platform in just a few moments

Implementation is done with just one line of simple code – call the function and ALL is done for you in the background – see code sample in appendix A

The API is very basic & simple SOAP (can of course also use XML)

ARKOM provides a FULL operative environment for Development & Tests as well as TEST TerminalNum & Test Password.

it is completely separated from the LIVE environment

Transactions executed on TEST system behaves exactly like LIVE except there will be no charge or any other operation the used card

The URL address for development and testing only is

[https://secure.arkom.co.il/wsdev/MTS\\_WebService.aspx](https://secure.arkom.co.il/wsdev/MTS_WebService.aspx)

LIVE URL will be provided by ARKOM once development & QA is done

#### **Note:**

- parameters names are case sensitive.
- Returning Error Description "**The remote server returned an error: (502) Bad Gateway**" means there is a missing mandatory parameter
- This document describes ONLY Terminal Based transactions (Card Present & Card Not Present) for other CNP Transactions (Internet etc) please refer to the full MTS API documentation

## API's exposed functions

Function name	Description/comments
MTS_mFlex_CreateVUID	Ask the Server for a VUID
MTS_mFlex_DoTransaction	Send a transaction to be executed by the Terminal
MTS_mFlex_CancelTransaction	Send a transaction CANCEL operation to the Terminal *1
MTS_mFlex_RetrieveTransactionByVUID	Retrieve transaction result from Server
MTS_mFlex_Transmit	Transmit & Deposit transactions file *3
MTS_mFlex_GetMerchantInfo	Retrieve Merchant info from Terminal *4
MTS_AshEMV_GetErrorDescription	Retrieve error description by error code
MTS_EMV_RetakeTrans_XML_Notes	Retrieve transaction's slips (Notes)
MTS_mFlex_DoSimpleTransaction	An easier lighter version of MTS_mFlex_DoTransaction (less params)
MTS_mFlex_TerminalDisplay	Displays 3 lines permanent message on screen (until device reset)

## Comments' remarks

1. The system supports a transaction cancellation operation.  
this means if a transaction was executed and the merchant wishes to cancel it - it can be easily performed using this function.  
the basic condition is that the transaction to be cancelled is not yet transmitted  
otherwise a full refund is required
2. Retrieving a transaction from the terminal is possible  
it will produce the same result as retrieving from server only a bit slower  
the basic condition is that the transaction to be retrieved is not yet transmitted
3. Transmit transactions file to the Credit Company  
Each Terminal should perform its own Transmitting. The host application should have functionality to perform on demand and / or automatically launch at specific time at night  
There will be no automatic Transmit at night invoked by the server
4. Retrieve Merchant ID & Name from Terminal  
Operative but Not really useful...

2

## Note:

Successful transaction's slip can be retrieved by using the function **MTS\_mFlex\_RetrieveTransactionByVUID** or by using **MTS\_EMV\_RetakeTrans\_XML\_Notes**.

**MTS\_EMV\_RetakeTrans\_XML\_Notes** can be used for BOTH Card Present & Card NOT Present transactions where **MTS\_mFlex\_RetrieveTransactionByVUID** can be used ONLY for Card Present transactions taken by the Flex Terminal using **MTS\_mFlex\_DoTransaction** or **MTS\_mFlex\_CancelTransaction**.



## Correct flow vis-à-vis the clearing interface – Basic Programmer's Guide

1. If VUID is needed in the function you should first ask the Server for a VUID
2. This VUID should be saved for transaction retrieval in case of any failure event.  
without VUID there is no way to retrieve the transaction result
3. Send your request to the MTS server and hold for the result to return
4. Every function has a time out managed by the server. It is recommended to maintain your own internal time out in case of deadlock or any other freeze issue or alternatively create an option to break waiting for the result (Minimum recommended TimeOut should be no less than 3 Minutes)
5. All functions return the full JSON in the results (where applicable & relevant) so you can process it as you wish- if needed

### Programmer note:

The programmer should pay attention to the function **return code** and **Status** (if applicable for example in transactions functions) to process the results

Function return code of Zero (0) means the **function** ended successfully **NOT** the transaction

A function can return Zero (0) and the transaction can still fail - so make sure both are checked

## Functions & Parameters

- Parameter marked in red indicates a parameter sent by reference to retrieve data from the function
- Parameter marked with \* indicated a mandatory field & value
- All functions return an integer.  
0 = OK, any other value means the function failed. Field RetStr will hold the error descriptions  
it is recommended to use MTS Error Description Retrieval by code to have the Error Description
- Return value of 0 from a transaction function **does not** mean the transaction succeeded, this is returned in field Status. If transaction failed the error returns in StatusMessage
- ALL functions require ALL parameters to be sent. Empty values should be initialized ("" ) not NULL

## MTS\_EMV\_RetakeTrans\_XML\_Notes

Parameter name	Description/comments
ARKOM_TerminalNumber*	Fixed - Provided by ARKOM
ARKOM_TerminalPassword*	Fixed - Provided by ARKOM
TransID (VUID)*	Transaction ID / VUID
ansXML	XML or JSON represents transaction result
mNote	Merchant's copy of Note
cNote	Customer's copy of Note

4

## MTS\_AshEMV\_GetErrorDescription

Parameter name	Description/comments
ARKOM_TerminalNumber*	Fixed - Provided by ARKOM
ARKOM_TerminalPassword*	Fixed - Provided by ARKOM
ErrorCode*	Error code as received from function result
ErrorDescription	Error Description

## MTS\_mFlex\_CreateVUID

Parameter name	Description/comments
ARKOM_TerminalNumber*	Fixed - Provided by ARKOM
ARKOM_TerminalPassword*	Fixed - Provided by ARKOM
VUID	Transaction ID
RetStr	Function return error message

MTS\_mFlex\_DoTransaction

MTS\_mFlex\_CancelTransaction \* (see remark)

MTS\_mFlex\_RetrieveTransactionByVUID \* (see remark)

Parameter name	Description/comments
ARKOM_TerminalNumber*	Fixed - Provided by ARKOM
ARKOM_TerminalPassword*	Fixed - Provided by ARKOM
TerminalID*	Fixed - Provided by ARKOM / Generated on Pairing by the Terminal
VUID*	Transaction ID – Received from MTS (by separate function)
EntryMode	See description table bellow
mNote	Merchant's copy of note – formatted & ready to dump
cNote	Customer's copy of note – formatted & ready to dump
Status	Transaction execution return status (see table below)
StatusMessage	Transaction execution return message
UID (Mandatory for Cancel Transaction)	SHVA UID (Trans ID for SHVA)
Mutag	See table bellow
Solek	See table bellow
Manpik	See table bellow
CardNumber	PAN (on merchant's copy – full. on customer's – masked)
TranType	See table bellow
Amount	Transaction amount (####0.00")
CreditTerms	See table bellow
Payments	If Installments or Credit transaction – number of payments
FirstPaymentAmount	Installments – amount of first payment. Leave empty for auto calc (####0.00")
OtherPaymentAmount	Amount of each other payment after the first (auto calc) (####0.00")
IssuerAuthNum	Issuer Authorization Number
AuthCodeManpik	Approval Authorization source – who approved the transaction
Currency	ILS=376 (default). Other currencies see ISO-4217
TranCode	See table bellow
CardName	Card name
ExpDate	Card Expiration date
AdditionalInfo	Free info sent to server – returns back ASIS
TransResultJSON	Full result JSON
RetStr	Function return error message

Remarks:



- When Cancelling a Transaction, the VUID must be a **NEW** VUID generated by the server and the UID must be the UID of the Transaction to cancel.  
That transaction should be alive on the terminal (not transmitted yet)  
All other transaction field can be left empty (but not NULL)
- To Retrieve Transaction By VUID the required Transaction's VUID must be provided to the function. All other parameters will be loaded with the retrieved transaction info

## MTS\_mFlex\_Transmit

Parameter name	Description/comments
ARKOM_TerminalNumber*	Fixed - Provided by ARKOM
ARKOM_TerminalPassword*	Fixed - Provided by ARKOM
TerminalID*	Fixed - Provided by ARKOM / Generated on Pairing by the Terminal
TransReport	Transactions Deposit Report (formatted ready to Dump)
TransAsmachta	Deposit Reference Number (Asmachta)
Status	Transaction execution return status (see table below)
StatusMessage	Transaction execution return message
ReturnJSON	Full result JSON
RetStr	Function return error message

## MTS\_mFlex\_GetMerchantInfo

Parameter name	Description/comments
ARKOM_TerminalNumber*	Fixed - Provided by ARKOM
ARKOM_TerminalPassword*	Fixed - Provided by ARKOM
TerminalID*	Fixed - Provided by ARKOM / Generated on Pairing by the Terminal
MerchantID	Merchant's ID coded into Terminal
MerchantName	Merchant's Name coded into Terminal
ReturnJSON	Full result JSON
Status	Transaction execution return status (see table below)
StatusMessage	Transaction execution return message
RetStr	Function return error message



## MTS\_mFlex\_DoSimpleTransaction

Parameter name	Description/comments
ARKOM_TerminalNumber*	Fixed - Provided by ARKOM
ARKOM_TerminalPassword*	Fixed - Provided by ARKOM
TerminalID*	Fixed - Provided by ARKOM / Generated on Pairing by the Terminal
VUID*	Transaction ID – Received from MTS (by separate function)
Amount	Transaction amount (####0.00")
Currency	Currency (leave empty for 376 = ILS as default)
EntryMode	EMV or empty for EMV (Chip & Pin) or Anything else for MOTO (Phone – CNP)
TranType	"C" for Charge or "R" for Refund
Status	Transaction execution return status (see table below)
StatusMessage	Transaction execution return message
mNote	Merchant's copy of slip
cNote	Customer's copy of slip
Additional Info	Private info sent to function (PrmX) is returned untouched
RetStr	Function return error message

In most cases this can be the function used to create a transaction

All is handled by the backend & the Terminal

Basically, it is meant for a Go / NoGo result without too much work around it

Should full transaction result be required or for a later control flow it is always possible to use

**MTS\_mFlex\_RetrieveTransactionByVUID** any time to retrieve full transaction's results structure

8

## MTS\_mFlex\_TerminalDisplay

Parameter name	Description/comments
ARKOM_TerminalNumber*	Fixed - Provided by ARKOM
ARKOM_TerminalPassword*	Fixed - Provided by ARKOM
TerminalID*	Fixed - Provided by ARKOM / Generated on Pairing by the Terminal
MsgLine1	First on-screen message line (Max 30 chars) will be centered
MsgLine2	Second on-screen message line (Max 30 chars) will be centered
MsgLine3	Third on-screen message line (Max 30 chars) will be centered
RetStr	Function return error message

## Codes / Parameters sent / returned in functions

TranCode	TranType

<ul style="list-style-type: none"> <li>1 - Normal</li> <li>2 - Pre Auth*</li> <li>3 - Pre-Auth Completion*</li> <li>4 - Pre-Auth Cancelation*</li> <li>5 - Cancel</li> <li>6 - Authorized Transaction*</li> </ul>		<ul style="list-style-type: none"> <li>1 - Regular</li> <li>3 - Forced Transaction*</li> <li>6 - Cash Back</li> <li>7 - Cash*</li> <li>30 - Balance</li> <li>53 - Refund</li> </ul>	
<p><b>EntryMode</b></p> <ul style="list-style-type: none"> <li>1 - MSR</li> <li>5 - Contact Less</li> <li>40 - Contact EMV</li> <li>50 - Phone</li> <li>51 - Signature only</li> </ul> <p>Leave BLANK for EMV defaults</p>		<p><b>CreditTerms</b></p> <ul style="list-style-type: none"> <li>1 - Regular</li> <li>2 - Special Credit</li> <li>3 - Immediate</li> <li>6 - Credit</li> <li>8 - Settlements</li> </ul>	
<p><b>Currency - ISO 4217 (samples)</b></p> <p>ILS   376   Israeli New Shekel</p> <p>USD   840   US Dollar</p> <p><b>Note:</b> Flex Terminal Supports Only ILS &amp; USD</p>		<p><b>Mutag</b></p> <ul style="list-style-type: none"> <li>0 - Private Label "כרטיס פרטי"</li> <li>1 - MasterCard "מסטוקארד"</li> <li>2 - Visa "ויזה"</li> <li>3 - Diners "דיינרס"</li> <li>4 - AmEx "אמקס"</li> <li>5 - Isracard "ישראלכרט"</li> <li>6 - "JCB"</li> <li>7 - "DISCOVER"</li> <li>8 - Maestro "מאסטרו"</li> </ul>	
<p><b>Solek (Acquirer)</b></p> <ul style="list-style-type: none"> <li>1 - Isracard "ישראלכרט"</li> <li>2 - ICC "באל"</li> <li>6 - MAX "מקס / לאומי קארד"</li> </ul>		<p><b>Manpik (Issuer)</b></p> <ul style="list-style-type: none"> <li>0 - Tiurist Card "תייר"</li> <li>1 - Isracard "ישראלכרט"</li> <li>2 - ICC "באל"</li> <li>6 - MAX "מקס / לאומי קארד"</li> </ul>	



Please note:

Please contact the support department to receive the terminal number and access password to the development system

This can be done directly by clicking on the following link [info@arkom.co.il](mailto:info@arkom.co.il)

Transactions sent in the test environment will not be charged to the cards

For the purposes of the tests and system development, the complete transaction procedure will be carried out, including the obtaining of approvals, but no charge or financial transaction will be made.

It is therefore possible to use real credit cards for the purpose of testing with no concern that charges will occur

## Appendix A – Implementation Code Sample

Implementation snippet for **MTS\_mFlex\_DoTransaction** function using VB.Net

```
Dim iRet As Integer = mtsWS.MTS_mFlex_DoTransaction(ARKOM_TerminalNum,
                                                    ARKOM_TerminalPassword,
                                                    TerminalID,
                                                    VUID,
                                                    EntryMode,
                                                    mNote,
                                                    cNote,
                                                    Status,
                                                    StatusMessage,
                                                    UID,
                                                    Mutag,
                                                    Solek,
                                                    Manpik,
                                                    CardNumber,
                                                    TranType,
                                                    Amount,
                                                    CreditTerms,
                                                    Payments,
                                                    FirstPaymentAmount,
                                                    OtherPaymentAmount,
                                                    IssuerAuthNum,
                                                    AuthCodeManpik,
                                                    Currency,
                                                    TranCode,
                                                    CardName,
                                                    ExpDate,
                                                    AdditionalInfo,
                                                    TransResultJSON,
                                                    sRet
                                                    )

If iRet = 0 Then
    'Function had succeeded
    If Status = "0" Then
        'Transaction succeeded
        'do your flow for a succeeded transaction
        'you get the Slips (Notes) formatted and ready to dump
        'all other params can be ignored or be processed according your design
    Else
        'Function Failed
        'do your transaction failure flow
        'you get Status & StatusMessage to describe the failure
    End If
Else
    'Function Failed
    'Function Failed - follow your function fail flow
    'iRet holds the return error code
    'use MTS_AshEMV_GetErrorDescription to retrieve error description
End If
```

11

Tip:

You can use **CreditTerms** as empty string ("") to have the terminal handle full flow including setting the transaction type (Regular/Credit/Installment – and installment info as well) and get it all ready in the answer received from the function, so no need to worry about internal data – let it all be handled by the function & Terminal – just send **Amount**, **TranType (1)**, & **TranCode (usually const = "1")**

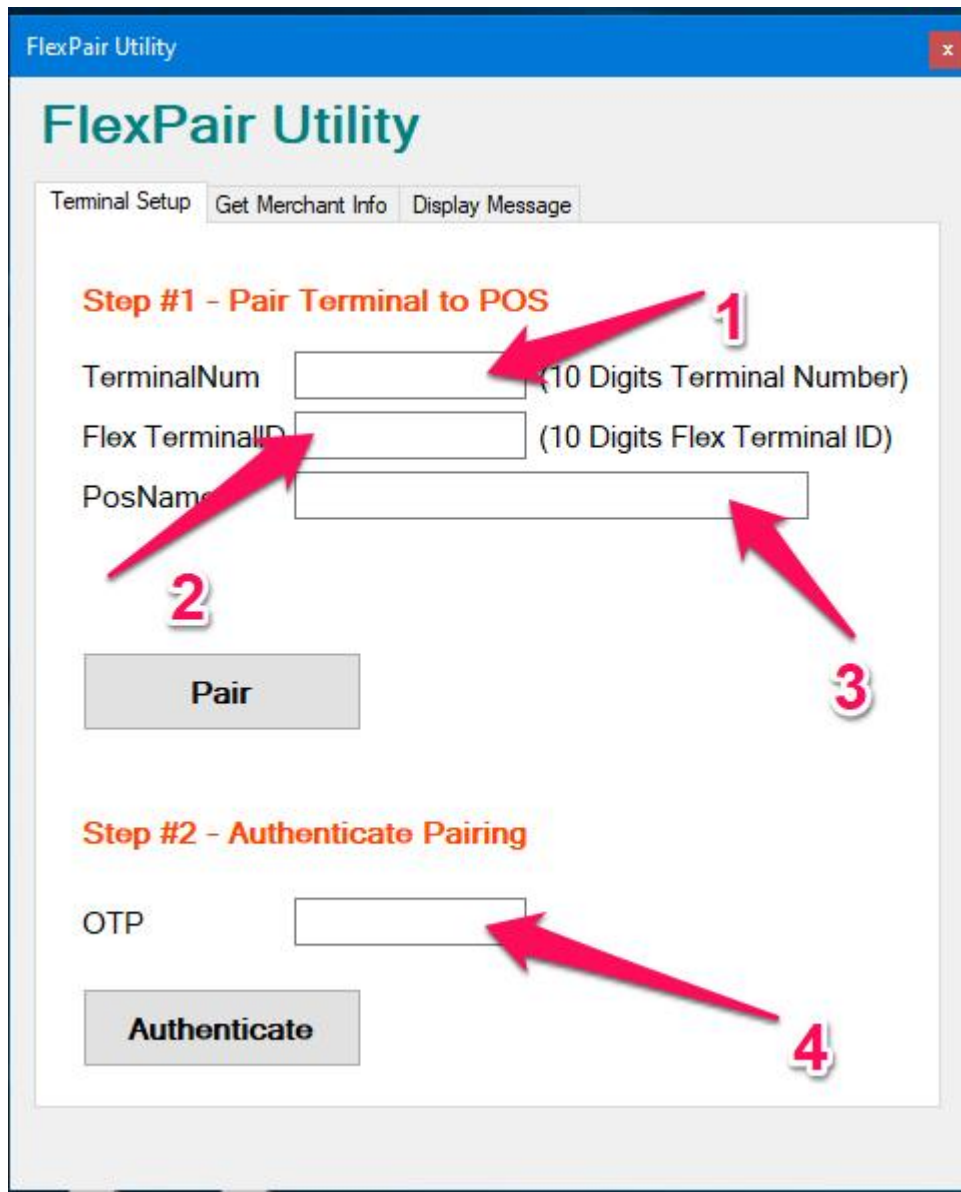
## Appendix B – Flex Terminal Pairing

**Each and Every FLEX Terminal MUST be Paired prior being usable**

The Pairing procedure is very simple yet requires attention

ARKOM provides a tiny utility that should be downloaded from our web site

The Pairing utility looks like the following picture & it takes 2 simple steps to pair a FLEX Terminal



The screenshot shows the 'FlexPair Utility' window with a blue header. Below the header, there are three tabs: 'Terminal Setup', 'Get Merchant Info', and 'Display Message'. The main content area is titled 'FlexPair Utility' and contains two steps:

**Step #1 - Pair Terminal to POS**

TerminalNum  (10 Digits Terminal Number) **1**

Flex TerminalID  (10 Digits Flex Terminal ID) **2**

PosName  **3**

**Pair**

**Step #2 - Authenticate Pairing**

OTP  **4**

**Authenticate**

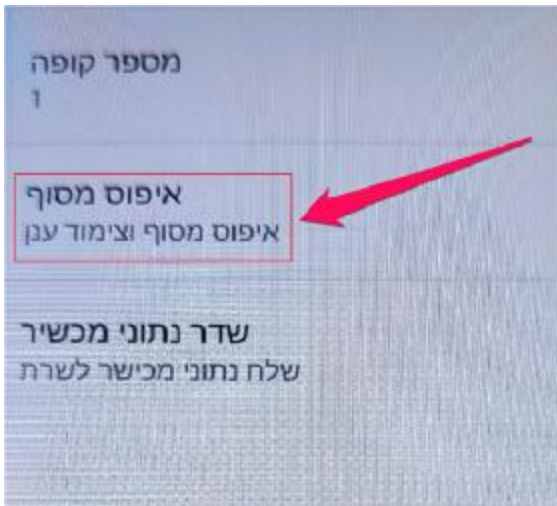
Red arrows point to the input fields for TerminalNum (1), Flex TerminalID (2), PosName (3), and OTP (4).



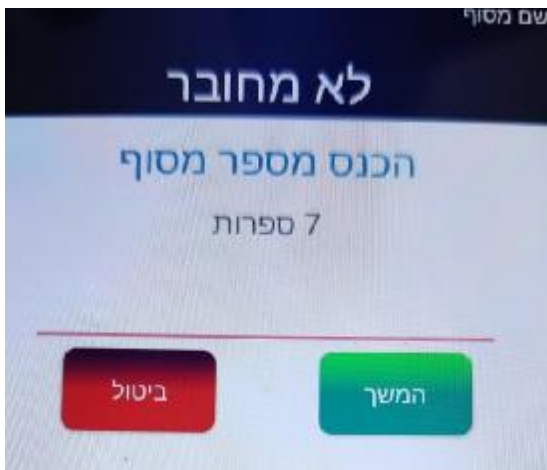
**Note:**

Before you can pair the terminal, it should be reset and a local terminal is to be set.

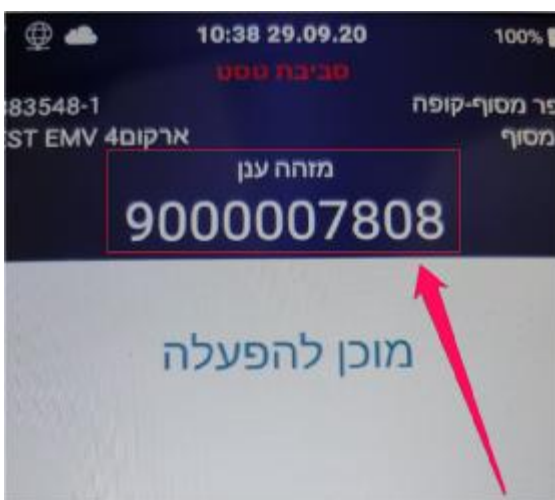
1. From the main terminal app screen use the hidden button to enter terminal's settings
2. Scroll down to "Terminal Reset" option
3. Once the terminal is reset return to the main terminal's app screen
4. You will be prompted to enter the Terminal Number (here you should key in the first 7 digits from 10 digits terminal number provided by ARKOM)
5. Once this is done just press OK and the terminal will initialize and download all required data
6. after this step is finished you should get a Terminal Cloud ID shown on top of the screen (see photo)



This is the Terminal's setting option to activate for terminal reset



This is the terminal's init screen – here you key in the 7 digits terminal number



This is the FLEX terminal's cloud ID

## Step #1: Start Pairing Procedure

FLEX Terminal unique identification data

1. Fill in the Terminal Number as provided by ARKOM  
this is a 10 digits numeric value (see step #1 on the photo)
2. Fill in the Terminal's Cloud ID as displayed on the terminal's screen  
this is a 10 digits numeric value (see step #2 on the photo)
3. Now fill in the terminal's POS Name  
this is a free Alpha Numeric (English/Hebrew/Digits) value  
(see step #3 on the photo)

15

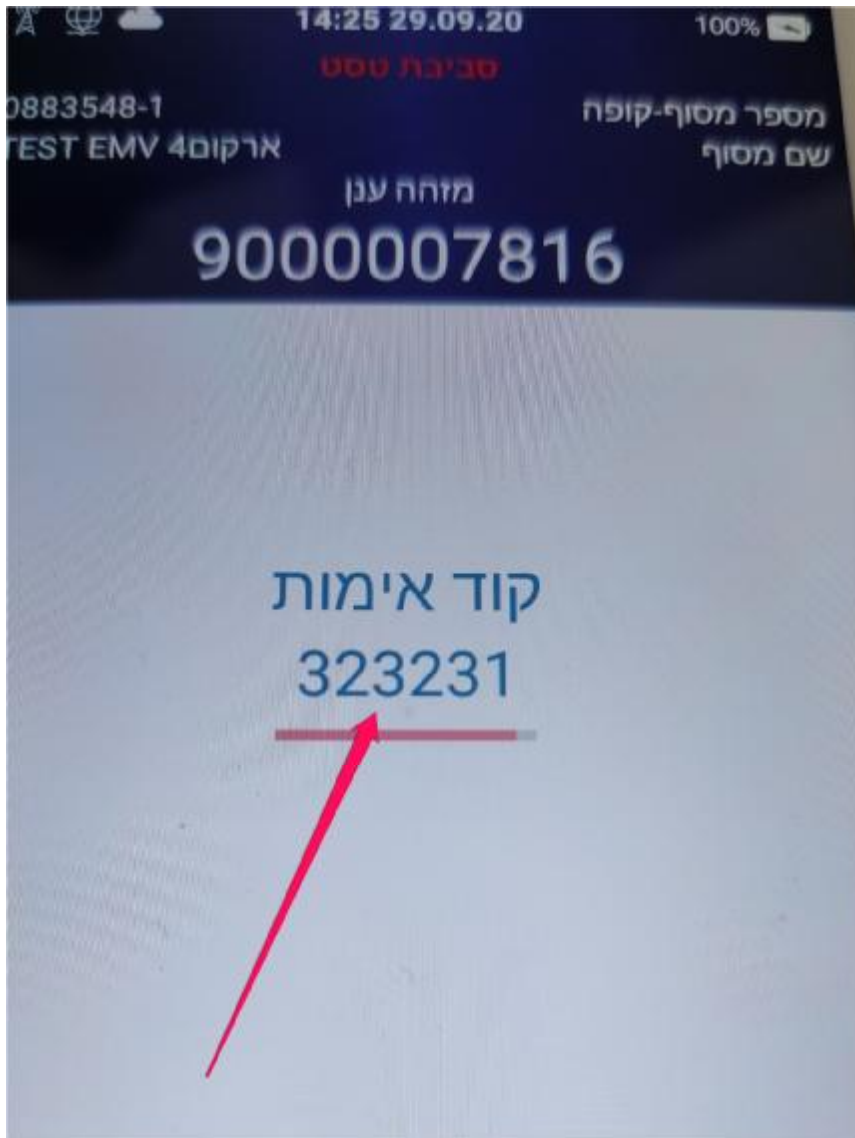
Once all required data fields are filled correctly click on [Pair] Button and then approve the notice



## Step #2: Complete Pairing Procedure

After a few seconds you should see an OTC code on the terminal's screen

This code will be valid for a short time and if not completed it will be vanished the Pairing procedure should be re-initialized



16

After completing this step the terminal is Paired and ready to use

### Note:

**ANY change to ANY terminal's parameter requires re-Pairing !!**