easypaisa

Easypaisa

Integration Guide

V.4.1

Contents

1. How to activate your sandbox test account:

Your Sandbox (testing) account will be created by your designated Telenor POC. You will receive an email from **ep.notification@telenor.com.pk** on your provided email ID containing the link to your Sandbox account activation along with the one time password sent on your registered number that you shared for sandbox creation.

Steps:

- Click the activation link in the email
- Enter your username. Your Username will be the email address provided by you for sandbox
- Enter OTP (one time password). Sent to you on your number
- Set a new password
  You're good to go!

Once you have successfully logged in to your Testing account. You will need two things in order to start your integration.

1-**Store ID**: To find your store ID, login to the OPS Portal and click on the profile button on top right

2-**Hashkey** (Used only in Post Method for Credit Card Transactions): To generate your Hashkey. Go to Account Settings>Generate Hashkey

2. Types of custom code Integration:

REDIRECTION	**API**
Post Method Integration only for Credit Card and Internet Banking Transactions	SOAP based OPEN API only for Mobile Account and Over the Counter Transactions

3. Post Method Integration for Credit Card Transactions

Post method is a seamless 2 step process

STEP 1: The merchant needs to POST below parameters in a form to Easypaisa on the f o l l o w i n g STAGING URL:

URL: https://easypaystg.easypaisa.com.pk/easypay/Index.jsf

Parameters Below:

Parameter	O/M	Name	Explanation	Possible Values
Amount	M	amount	Total amount of the transaction in PKR	Numeric, kindly set the amount to one decimal point e.g. 10.0
Store Id	M	storeId	Id of the store as provided by the	Numeric
Post back URL	M	postBackURL	The first post back URL for confirmation	Character. This should be a public URL
Order Reference Number	M	orderRefNum	Merchant Generated order reference number	Character
Merchant Hashed	M	merchantHashedReq	Hash value. Explained in detail in section 3.3	Character
Payment Method	M	paymentMethod	Merchant will redirect their customers to Easypaisa secure checkout for CC/IB transactions	IB_PAYMENT_METHOD CC_PAYMENT_METHOD
Expiry Date	O	expiryDate	Merchant provided expiry date for the particular transaction	YYYYMMDD HHMMSS
Auto Redirect	O	autoRedirect	If merchant wants to redirect to final post back URL and the end of transaction	0/1
Email Address	O	emailAddr	If the merchant wants to pass the customer's entered email address it would be pre populated on Easypaisa	Example value Test@abcd.com
Mobile Number	O	mobileNum	If the merchant wants to pass the customer's entered mobile number it would be pre populated on Easypaisa checkout screen.	Example Value 03321234567

Bank Identifier	O	bankIdentifier	If the merchant wants to pass the customer's selected Bank (Identification Number) in order to	6 digit alphanumeric value

2. After completing the form in Step 1 the customer will be pressing the Proceed Button and lands back on the merchant website on the same URL given in postbackURL variable in the first step. This will be a confirmation screen on merchant's website to perform a handshake between Easypaisa and merchant's website. Then Easypaisa sends back a parameter named auth_token to the postbackURL which is sent as a GET parameter. Now the merchant needs to post back below two parameters again to the following URL:

   URL: https://easypaystg.easypaisa.com.pk/easypay/Confirm.jsf

   Parameters:

   auth_token

   po stBackURL

After this redirection the Easypaisa authenticates the auth_token sent by merchant with t h e one it has in the previous step, and upon successful authentication it will make customer land on the successful checkout screen sending back following three variables to the second postBackURL:

status

desc

orderRefNumber

Parameter	O/M	Name	Explanation	Possible Values
Status	M	status	Status of the transaction request made by the merchant	"Success", "Failure"
Description	M	desc	Code ID for the status	"0000", "0001"
Order Reference Number	M	orderRefNum	Merchant Generated order reference number	Character

3.1 Post Method Sample Codes

**Below are the only working codes that we have for post method integration. In case your platform is developed on some other language then reference codes are available and will be provided on request, but you will have to customize and develop them as per need.

PHP Sample Code:

https://www.dropbox.com/s/sl1ovs4594dlfxk/Post%20Method%20PHP%20Sample%20Code.txt?dl=0

.Net Sample Code:

https://www.dropbox.com/s/qfxvr2rm2m298hp/Post%20Metod%20.NET%20Smaple%20Code.txt?dl=0

3.3 Encryption Algorithm to get value of "merchantHashedReq" Parameter:

In order to mitigate parameter tempering/modification while transfer and posting of data, merchant can encrypt the request using the hash key generated from your sandbox account. In your request to our online payment system you have to send ["merchantHashedReq"] parameter. Value for this parameter can be formed by encrypting your other parameters. Kindly note that all the parameters you were sending previously will be used in generation of the encrypted value. Moreover, correct sequence of parameters in the concatenated string should be like this:

amount=&autoRedirect=&emailAddr=&mobileNum=&orderRefNum=&paymentMethod=&postBackURL=&storeId=

The secret key to encrypt values can be obtained from Merchant Portal [Account Settings -> Generate Hash Key].

This encrypted request is sent along with the main request, which is then reconciled at OPS end to detect if parameter is changed or not. The encryption can be done using following algorithm:

1. Create map of all the fields that are part of the request:

```
Map<String, String> fields = new HashMap<String, String>();

fields.put("amount", "10.0");
fields.put("storeId", "28");
fields.put("orderRefNum", "11001");
fields.put("expiryDate", "20150101 151515");

fields.put("postBackURL", "http://localhost:9081/local/status.php");
```

2. Get the list of field name from the map created in the first step

```
List fieldNames = new ArrayList(fields.keySet());
```

3. Sort the map fields based on map key in alphabetical order

```
Collections.sort(fieldNames)
```

4. Create a string in following format. Remember that Auto redirect is a mandatory parameter while creating this string

amount=10.0&autoredirect=0&expiryDate=20150101151515&orderRefNum=11001&postBackURL=http://localhost:9081/local/status.php&storeId=28

5. Use AES/ECB/PKCS5Padding algorithm to encrypt with the key and string produced in the previous step

```
Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");  SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), "AES");  cipher.init(Cipher.ENCRYPT_MODE, secretKey);  encryptedValue = new String(Base64.encodeBase64(cipher.doFinal(value.getBytes())));
```

3.4 Hash Encryption Sample Code:

Merchants can use the below given PHP code as a reference for encryption process in Post Method. For any other language, please follow this reference code to create your own code as per need

Hash Encryption PHP Sample Code:

https://www.dropbox.com/s/eltjz30qdoycuoj/Hash%20encryption%20PHP%20Code.txt?dl=0

4. Open API for Mobile Account & Over The Counter Transactions:

Easypaisa provides the capability for B2B integrations by exposing core services to merchants to reuse their existing interfaces in order to integrate with Easypaisa. There will be no redirection to Easypaisa checkout page and merchants system will directly invoke Easypaisa APIs for initiating and inquiring the transaction.

Following two APIs are in-scope:

1. **Initiate a Transaction**
2. **Inquire Transaction Status**

The communication protocol supported is SOAP over HTTPs. Merchant's Systems should have the capability to perform SSL based communication with the Easypaisa Load Balancer.

The web service may be integrated using the following link:

https://easypaystg.easypaisa.com.pk/easypay-service/PartnerBusinessService/META-INF/wsdl/partner/transaction/PartnerBusinessService.wsdl

4.1 Integrate via Open API

Following are the steps that are required for integration via Open API:

1. Generate a WSDL client using the WSDL URL

(https://easypaystg.easypaisa.com.pk/easypay-service/PartnerBusinessService/META-INF/wsdl/partner/transaction/PartnerBusinessService.wsdl)

2. After the client generation two SOAP operations would be exposed as follows:

    a. initiate Transaction – This operation can be used to initiate OTC and MA transactions only.

    b. inquire Transaction – This operation can be used to inquire about the status of any transaction.

Testing Open API Operations

SOAP UI tool v.5 or above can be used in order to test open APIs.

4.2 Initiate Transaction:

Request Parameters:

Field Name	Description	Mandatory/Optional	Data
Username	This will be provided by Easypaisa using "ManagePartner Accounts" screen. Provided by Easypaisa POC	M	String
password	Encrypted password. Sent to Merchants Email Address	M	String
channel	Channel through which transaction is initiated. (E.g. Internet, JPOS Device, etc.)		String
orderId	Merchant's system generated Order Id	M	String
storeId	Store ID generated during merchant registration in	M	Integer
transactionAmount	Total Transaction Amount	M	String
transctionType	Type of transaction. Possible values are: OTC, MA	M	String
msisdn	Customer's MSISDN	M in case of OTC O in case of MA	String
mobileAccountNo	Customer's Mobile Account #	O in case OTC M in case of MA	String
emailAddress	Customer's Email Address	O	String

Response Parameters:

Field Name	Description	Data Type
responseCode	Easypaisa generated response code. Possible values are: 0000 - Success 0001 - System Error 0002 - Required field is missing 0003 - Invalid Order ID 0004 - Merchant Account does not exist	String
orderId	Merchant's system generated Order Id	String
storeId	Store ID generated during merchant registration in	Integer
paymentToken	Token generated in case of OTC	String
transactionId	Transaction Id of FUNDAMO System for MA transactions only.	String
transactionDateTime	Transaction Date Time	Datetime
paymentTokenExiryDateTime	Token Expiry Date time	Datetime

Sample Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:dto="http://dto.transaction.partner.pg.systems.com/" xmlns:dto1="http://dto.common.pg.systems.com/">

    <soapenv:Header />

    <soapenv:Body>

        <dto:initiateTransactionRequestType>
            <dto1:username>pg-
            systems</dto1:Username>

            <dto1:password>9b01234324vxddc0b</dto1:Password>

            <channel>Internet</channel>

            <orderId>000000001</orderId><storeId>465</storeId>

            <transactionAmount>5000.00</transactionAmount>
```

```
                    <transactionType>OTC</transactionType>
                    <msisdn></msisdn>
                    <mobileAccountNo>034632401722</mobileAccountNo>
                    <emailAddress>abc@test.com</emailAddress>
            </dto:initiateTransactionRequestTyp
                                e>
    </soapenv:Body>
</soapenv:Envelope>
```

Sample Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
        <soapenv:Body>
                <ns3:initiateTransactionResponseType
                xmlns:ns2="http://dto.common.pg.systems.com/"
   xmlns:ns3="http://dto.transaction.partner.pg.systems.com/">
                        <ns2:responseCode>0000</ns2:responseCode>
                        <orderId>000000001</orderId>
                        <storeId>465</storeId>
                        <paymentToken>064463</paymentToken><transactionDateTime>2014-04-
                        25T19:59:37.981+05:00</transactionDateTime> <paymentTokenExiryDateTime>2014-
                        04-
                        26T19:59:37.968+05:00</paymentTokenExiryDateTime>
                </ns3:initiateTransactionResponseType>
        </soapenv:Body>
   </soapenv:Envelope>
```

4.3 Inquire Transaction

Request Parameters

Field Name	Description	Mandatory (M)	Data Type
username	This will be provided by Easypaisa using “Manage Partner	M	String
password	Encrypted password generated by “Manage Partner	M	String
orderId	Merchant’s system generated Order Id	M	String
accountNum	Merchant’s Account Number registered with Easypaisa	M	String

Response Parameters

Field Name	Description	Data Type
responseCode	Easypay generated response code. Possible values are: 0000 - Success 0001 - System Error 0002 - Required field is missing 0003 - Invalid Order ID	
orderId	Merchant’s system generated Order Id	String
storeId	Merchant Store ID #	Integer
accountNum	Merchant Account No registered with Easypaisa	String
storeName	Merchant Store Name	String
paymentToken	Token generated in case of OTC	String
transactionId	Easypaisa generated unique Transaction Id	String
transactionStatus	Transaction Status possible Values are: REVERSED, PAID,	String
transactionAmount	Total transaction Amount	Double
transactionDateTime	Transaction Datetime	Datetime
paymentTokenExiryDateTime	Token expiration date time in case of OTC	Datetime
transactionPaidDateTime	Transaction Paid Date Time	Datetime
Msisdn	Customer MSISDN	String
paymentMode	Mode of payment (OTC, MA)	TransactionType

Sample Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:dto="http://dto.transaction.partner.pg.systems.com/" xmlns:dto1="http://dto.common.pg.systems.com/">

    <soapenv:Header />

    <soapenv:Body>

        <dto:inquireTransactionRequestType>
            <dto1:username>pg-
            systems</dto1:Username>

            <dto1:password>9b01234324vxddc0b</dto1:Password>

            <orderId>0000001</orderId>

            <accountNum>9999999999999999</accountNum>

        </dto:inquireTransactionRequestType>

    </soapenv:Body>

</soapenv:Envelope>
```

Sample Response

```
                    <soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

  <soapenv:Body>

        <ns3:inquireTransactionResponseType

          xmlns:ns2="http://dto.common.pg.systems.com/"      xmlns:ns3="http://dto.transaction.partner.pg.systems.com/">

            <ns2:responseCode>0000</ns2:responseCode>

            <orderId>000001</orderId>

            <accountNum>9999999999999999</accountNum>

            <storeId>1001</storeId> <storeName>Test

            Store</storeName>

            <paymentToken>668551</paymentToken>

            <transactionId>100591</transactionId>
```

<transactionStatus>PAID</transactionStatus>

<transactionAmount>10000.00</transactionAmount><transactionDateTime>2014-04-25T19:59:37.981+05:00</transactionDateTime> <paymentTokenExiryDateTime>2014-04-25T19:59:37.981+05:00</paymentTokenExiryDateTime><transactionPaidDateTime>2014-04-25T19:59:37.981+05:00</transactionPaidDateTime> <paymentTokenExiryDateTime>2014-04-26T19:59:37.968+05:00</paymentTokenExiryDateTime><msisdn>03463240172</msisdn>

<paymentMode>OTC</paymentMode>

</ns3:inquireTransactionResponseType>

</soapenv:Body>

</soapenv:Envelope>

5. Instant Payment Notification:

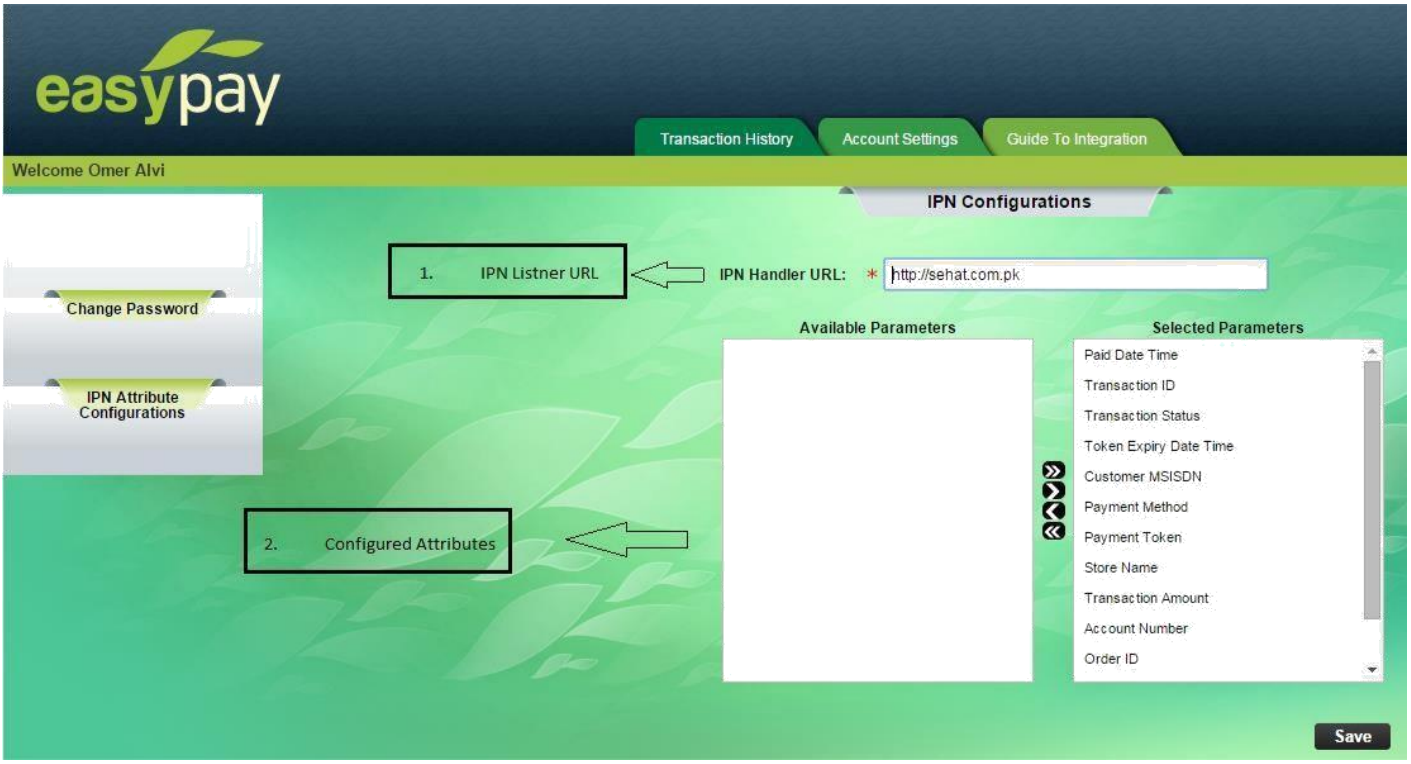
Instant payment notification message is used to notify merchants about the details of any particular transaction made by customers using Easypaisa channel. IPN message is self-configured in Easypaisa for each merchant. It is customized as to what details merchant requires in response for any particular transaction. The IPN message service sends merchant a notification whenever Easypaisa transaction is created (with'Paid' status) or is updated to 'Paid'.

IPN Handler URL is the Merchant's IPN listener URL where Merchant will be expecting the response from Easypaisa. IPN message will contain the selected parameters configured for merchant using below mentioned screen in merchant portal.

5.1 What you need to do:

1. Configure the listener URL on Easypaisa merchant portal by going to the account setting tab and clicking on the IPN attributes configuration
2. Select desired IPN attributes from the Easypaisa merchant portal
3. Create a listener that will receive in GET request variable named 'url'
4. The received URL would be that of REST API which will return the IPN attributes

**See reference screenshot below



Following is the format for URL:

MerchantURL?url=**Rest API URL/Merchant Account ID/Order ID**

Where url = is the reserved word added by the system.

For Example:

Merchant URL = http://www.TestMerchant.com

Rest API URL = https://easypay.easypaisa.com.pk/easypay-service/rest/v1/order- status Merchant Account ID = 00001
OrderID = 998877

IPN Message:https://www.TestMerchant.com?url=
https://easypaystg.easypaisa.com.pk/easypay-service/rest/v1/order- status/1000223344/998877

6. Testing Credentials for Sandbox (Test Account)

Once the integration with Easypaisa is completed, merchants can test the payment flows using the following Credentials for Mobile Account and Credit Card Transactions.

**Each case must be tested as per screenshot below. There are a total of seven cases that are must to be tested. Without which merchant cannot go live.

For Easypaisa Shop payment method testing:

1- Generate an order
2- Pay with Easypaisa (Shop payment method)
3- Fill details on Easypaisa payment page
4- Copy the token number that has been generated
5- On the same page where token number is displayed, you will see an option 'Pay Token', click that option and paste the token number in it and click enter
6- Transaction should be marked as paid in Easypaisa portal

7. How to Go Live with your Easypaisa Account:

Once all the payment methods are tested on sandbox, you can then move to production (live account) if it has been created (depending on your account opening documentation status at Telenor Bank).

Live Account Activation

Your Live account will be created by your designated Telenor POC. You will receive an email from **ep.notification@telenor.com.pk** on your provided email ID containing the link to your live account activation along with the one time password sent on your registered number that you shared for sandbox creation.

Steps:

- Click the activation link in the email
- Enter your username. Your Username will be the email address provided by you for live account
- Enter OTP (one time password). Sent to you on your number
- Set a new password
  You're good to go!

Once you have successfully logged in to your live account. You will need two things in order to move from Staging to Live.

1-**Store ID**: To find your store ID, login to the OPS Portal and click on the profile button on top right

2-**Hashkey** (Used only in Post Method for Credit Card Transactions): To generate your Hashkey. Go to Account Settings>Generate Hashkey

Changes Required for POST Method Integration (to move to live):

Kindly follow the below mentioned steps in order to move from testing to live Easypaisa environment if you are using the POST method integration

- Change your Store ID from staging with the one you will get in your live account
- Generate Hashkey from your live account .To generates your Hashkey. Go to Account Settings>Generate Hashkey and replace it with the one you used for hash encryption in staging environment.
- Change the index.jsf and confirm.jsf URL's with the one's given below

https://easypay.easypaisa.com.pk/easypay/Index.jsf

https://easypay.easypaisa.com.pk/easypay/Confirm.jsf

Changes required for OPEN API integration:

Kindly follow the below mentioned steps in order to move from testing to live Easypaisa environment if you are using the OPEN API integration method.

- Simple change the SOAP based WSDL URL of API to the one provided below (https://easypay.easypaisa.com.pk/easypay-service/PartnerBusinessService/METAINF/wsdl/partner/transaction/PartnerBusinessService.wsdl)
- Your SOAP credentials for live account (Username & Password) will be shared with you by your Easy pay POC