easypaisa

ONLINE

Easypaisa

Integration Guide

V.4.1

Table of 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

Post Method Integration only for Credit Card Transactions

REST API

REST based 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 following 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 Transactions	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 the 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/eltjz30qdoycuoj/Hash%20encryption%20PHP%20Code.txt?dl=0

.Net Sample Code:

https://drive.google.com/file/d/0B1fKYKZTIIt5Z3c4NFlXbno3Q28/view

3.2 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:90 81/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())));
```

3.3 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. REST APIs

4.1 Purpose

Payment Gateway provides the capability for B2B integrations by exposing core services to external partners. This enables external partners to reuse their existing interfaces in order to integrate with Easypaisa. There will be no re-direction to Easypaisa checkout page and external partner’s system will directly invoke Easypaisa APIs for initiating and inquiring the transaction.

Following three APIs are in-scope:

1 Initiate OTC Transaction
2 Initiate MA Transaction
3 Inquire Transaction Status

The communication protocol supported is REST over HTTPs. External Systems should have the capability to perform SSL based communication with the Easypaisa Load Balancer.

4.2 Initiate OTC Transaction

The Initiate OTC Transaction method is used to create OTC transactions in OPS

Request URL:

Staging:

https://easypaystg.easypaisa.com.pk/easypay-service/rest/v4/initiate-otc-transaction

Production:

https://easypay.easypaisa.com.pk/easypay-service/rest/v4/initiate-otc-transaction

Request Header:

Header key	Header value	Description
Credentials	Base64Encoded username:password	Partner Account username & Password for authentication. They will be provided by Easypaisa team and the merchant will have to encode them to Base64 before including them in the request

Request Parameters:

Field Name	Description	Mandatory (M) / Optional (O)	Data Type
orderId	Merchant's system generated Order ID	M	String
storeId	Store ID It should be the same id which is associated with partner Account credentials sent in the Header	M	Long
transactionAmount	Total transaction amount	M	Double
transactionType	Type of transaction. Value : OTC	M	String
Msisdn	Format: 03xxxxxxxxx (11 - digits)	M	String
emailAddress	Format : abc@xyz.com	M	String
tokenExpiry	Format : yyyymmdd HHmmss	M	String
optional1		O	String
optional2		O	String
optional3		O	String
optional4		O	String
optional5		O	String

Response Parameters:

Field Name	Description	Data Type
orderId	Merchant’s system generated Order ID	String
storeId	Store ID generated during merchant registration in Easypaisa	Long
paymentToken	Token generated in case of OTC	String
transactionDateTime	Format : dd/MM/yyyy hh:mm a	String
paymentTokenExpiryDateTime	Format : dd/MM/yyyy hh:mm a	String
optional1		String
optional2		String
optional3		String
optional4		String
optional5		String
responseCode	Easypaisa generated response	String
responseDesc	Possible values are : Code : Description 0000 = SUCCESS 0001 = SYSTEM ERROR 0002 = REQUIRED FIELD MISSING 0005 = MERCHANT ACCOUNT NOT ACTIVE 0006 = INVALID STORE ID 0007 = STORE NOT ACTIVE 0008 = PAYMENT METHOD NOT ENABLED 0010 = INVALID CREDENTIALS 0015 = INVALID_TOKEN_EXPIRY 0016 = Expiry date should be future date	String

Sample Request:

```
{
    "orderId": "abc123",
    "storeId": "43",
    "transactionAmount": "1.23",
    "transactionType": "OTC",
    "msisdn": "03458508726",
    "emailAddress": "testEmail@gmail.com",
    "tokenExpiry": "20190723 232722"
}
```

Sample Response:

```
{
    "orderId": "abc123",
    "storeId": 43,
    "paymentToken": "40933012",
    "transactionDateTime": "11/08/2018 10:41 PM",
    "paymentTokenExpiryDateTime": "23/07/2019 11:27 PM",
    "responseCode": "0000",
    "responseDesc": "SUCCESS"
}
```

4.3 Initiate MA Transaction

The Initiate MA Transaction method is used to create MA transactions in OPS

Request URL:

Staging:

https://easypaystg.easypaisa.com.pk/easypay-service/rest/v4/initiate-ma-transaction

Production:

https://easypay.easypaisa.com.pk/easypay-service/rest/v4/initiate-ma-transaction

Request Header:

Header key	Header value	Description
Credentials	Base64Encoded username:password	Partner Account username & Password, for authentication. They will be provided by Easypaisa team and the merchant will have to encode them to Base64 before including them in the request

Request Parameters:

Field Name	Description	Mandatory (M) / Optional (O)	Data Type
orderId	Merchant's system generated Order ID	M	String
storeId	Store ID It should be the same id which is associated with partner Account credentials sent in the header	M	Long
transactionAmount	Total transaction amount	M	
transactionType	Value : MA	M	String
mobileAccountNo	Format: 03xxxxxxxxx (11 - digits)	M	
emailAddress	Format : abc@xyz.com	M	String
optional1		O	String
optional2		O	String
optional3		O	String
optional4		O	String
optional5		O	String

Response Parameters:

Field Name	Description	Data Type
orderId	Merchant’s system generated Order ID	String
storeId	Store ID generated during merchant registration in Easypaisa	Long
transactionId	Transaction ID of Ericsson (EWP ID)	String
transactionDateTime	Format : dd/MM/yyyy hh:mm a	String
optional1		String
optional2		String
optional3		String
optional4		String
optional5		String
responseCode	Easypaisa generated response	String
responseDesc	Possible values are : Code : Description 0000 = SUCCESS 0001 = SYSTEM ERROR 0002 = REQUIRED FIELD MISSING 0005 = MERCHANT ACCOUNT NOT ACTIVE 0006 = INVALID STORE ID 0007 = STORE NOT ACTIVE 0008 = PAYMENT METHOD NOT ENABLED 0010 = INVALID CREDENTIALS 0013 = LOW BALANCE 0014 = ACCOUNT DOES NOT EXIST	String

Sample Request:

```
{
    "orderId": "abc123",
    "storeId": "43",
    "transactionAmount": "1.23",
    "transactionType": "MA",
    "mobileAccountNo": "03458508726",
    "emailAddress": "testEnmail@gmail.com"
}
```

Sample Response:

```
{
    "orderId": "abc123",
    "storeId": 43,
    "transactionId": "253184",
    "transactionDateTime": "11/08/2018 11:30 PM",
    "responseCode": "0000",
    "responseDesc": "SUCCESS"
}
```

4.4 Inquire Transaction Status

The Inquire Transaction Status API is used to inquire transaction information of the referenced transaction.

Request URL:

Staging:

https://easypaystg.easypaisa.com.pk/easypay-service/rest/v4/inquire-transaction

Production:

https://easypay.easypaisa.com.pk/easypay-service/rest/v4/inquire-transaction

Request Header:

Header key	Header value	Description
Credentials	Base64Encoded username:password	Partner Account username & Password for authentication. They will be provided by Easypaisa team and the merchant will have to encode them to Base64 before including them in the request

Request Parameters:

Field Name	Description	Mandatory (M) / Optional (O)	Data Type
orderId	Merchant’s system generated Order ID as stored in OPS at the time of transaction initiation	M	String
storeId	Store ID .It should be the same id which is associated with partner Account credentials sent in the header	M	Long
accountNum	Merchant’s EWP Account Number present at Profile Page of Easypaisa Merchant portal **“EWP Account # “**	M	String

Response Parameters:

Field Name	Description	Data Type
orderId	Merchant’s system generated Order ID	String
accountNum	Merchant’s EWP Account Number	String
storeId	Store ID generated during merchant registration in Easypaisa	Long
storeName	Store Name	String
paymentToken	Token generated in case of OTC	String
transactionStatus	Transaction status Possible values are : PAID, FAILED, PENDING, BLOCKED , EXPIRED , REVERSED etc.	String
transactionAmount	Total transaction amount	Double
transactionDateTime	Format : dd/MM/yyyy hh:mm a	String
paymentTokenExpiryDateTime	Only for OTC Format : dd/MM/yyyy hh:mm a	String
Msisdn	Customer MSISDN	String
paymentMode	Type of transaction . Value : MA , OTC , CC	String
optional1		String
optional2		String
optional3		String
optional4		String
optional5		String
responseCode	Easypaisa generated response Possible values are : Code : Description 0000 = SUCCESS 0001 = SYSTEM ERROR 0002 = REQUIRED FIELD MISSING 0003 = INVALID ORDER ID 0004 = INVALID MERCHANT ACCOUNT NUMBER 0005 = MERCHANT ACCOUNT NOT ACTIVE 0006 = INVALID STORE ID 0007 = STORE NOT ACTIVE 0010 = INVALID CREDENTIALS	String
responseDesc		String

Sample Request:

```
    {
        "orderId " : "MS5007",
        "storeId" : "43",
        "accountNum" : "654123987"
    }
```

Sample Response

```
  {
      "orderId”: "MS5007",
      "accountNum" : "654123987",
      "storeId" : 43,
      "storeName" : "PG Store 1",
      "paymentToken" : "40931912",
      "transactionStatus" : "PENDING",
      "transactionAmount" : 12,
      "transactionDateTime" : "09/08/2018 10:04 PM",
      "paymentTokenExpiryDateTime" : "09/07/2019 05:06 PM",
      "msisdn" : "03458508726",
      "paymentMode" : "OTC",
      "responseCode " : "0000",
      "responseDesc" : "SUCCESS"
}
```

4.5 Possible Reasons of API Response:

Required Field Missing

Any Mandatory field is missing.

Incorrect email format in initiate MA or OTC requests.

Incorrect MSISDN format in initiate OTC request.

Incorrect mobile Account No format in initiate MA request.

Transaction amount is sent less than or equal to Zero in the initiate MA or OTC requests.

4.6 Optional Parameters

These parameters are present in the requests and responses of all three APIs mentioned above and are optional in each case.

If these attributes are provided in the request, then OPS will save the values of these parameters in OPS database against the transaction and return them in the response.

These parameters are available for selection in the 'IPN Attribute Configurations' screen in the Account Settings tab of the Merchant Portal.

If selected, these optional parameters will also be sent in the merchants' IPN to the configured URL.

5. Instant Payment Notification:

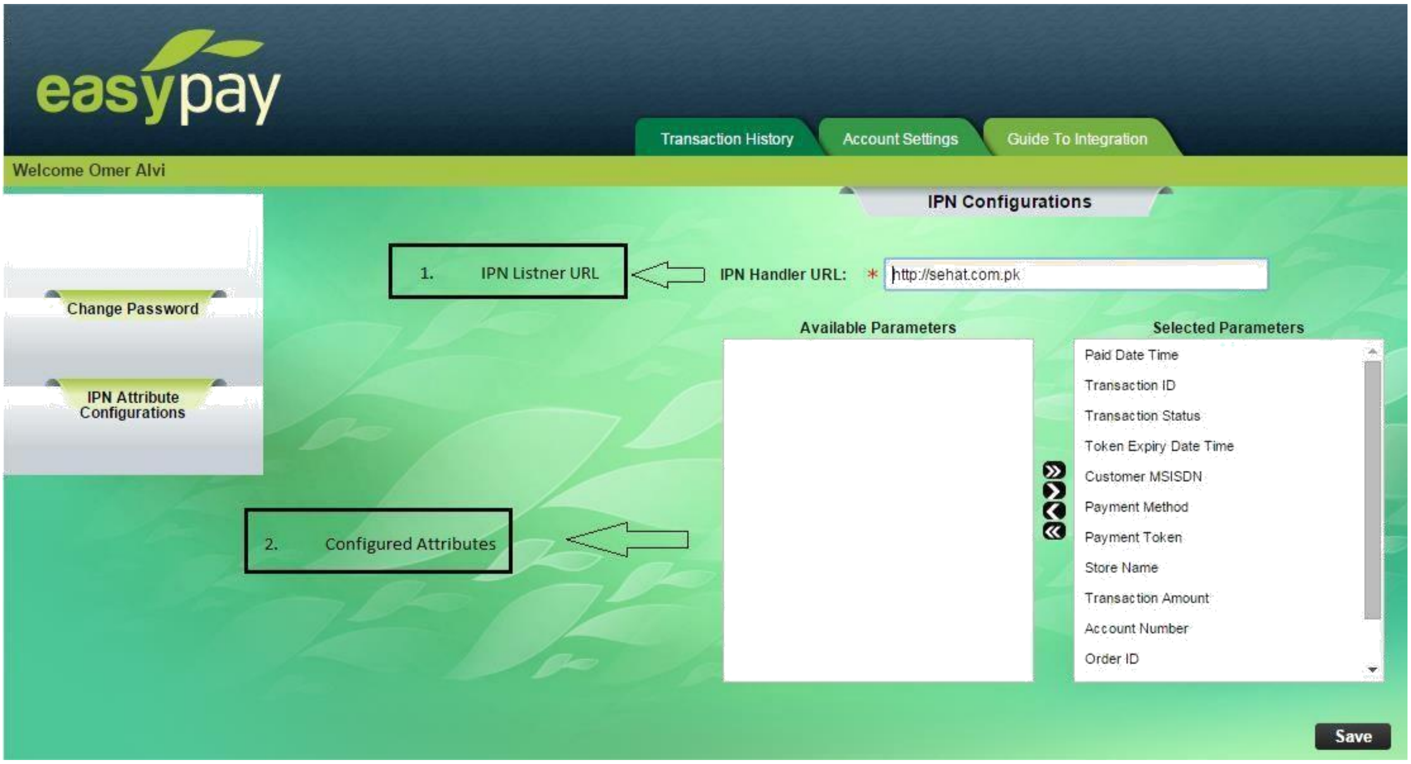
Instant payment notification message is used to notify merchants about the details of any particular transaction made by customers using Easypaisa channel. IPN message is self-configured in Easypaisa for each merchant. It is customized as to what details merchant requires in response for any particular 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



1. IPN Listner URL

IPN Handler URL: * http://sehat.com.pk

2. Configured Attributes

Selected Parameters:

- Paid Date Time
- Transaction ID
- Transaction Status
- Token Expiry Date Time
- Customer MSISDN
- Payment Method
- Payment Token
- Store Name
- Transaction Amount
- Account Number
- Order ID

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

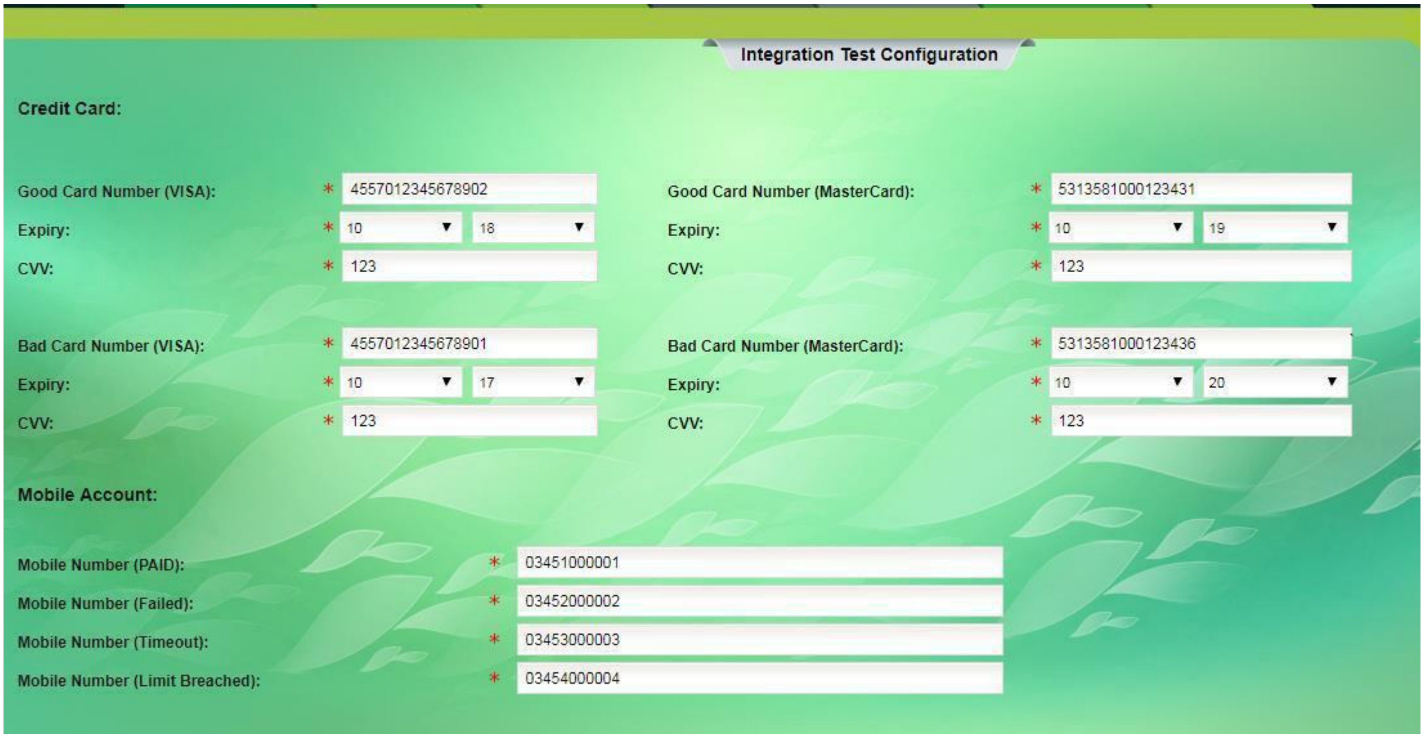
Rest API URL = https://easypay.easypaisa.com.pk/easypay-service/rest/v1/order- status Merchant Account ID = 00001 Order ID = 998877

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.



Integration Test Configuration

Credit Card:

Good Card Number (VISA):	* 4557012345678902		Good Card Number (MasterCard):	* 5313581000123431	
Expiry:	* 10 ▼	18 ▼	Expiry:	* 10 ▼	19 ▼
CVV:	* 123		CVV:	* 123	
Bad Card Number (VISA):	* 4557012345678901		Bad Card Number (MasterCard):	* 5313581000123436	
Expiry:	* 10 ▼	17 ▼	Expiry:	* 10 ▼	20 ▼
CVV:	* 123		CVV:	* 123	

Mobile Account:

Mobile Number (PAID):	* 03451000001
Mobile Number (Failed):	* 03452000002
Mobile Number (Timeout):	* 03453000003
Mobile Number (Limit Breached):	* 03454000004

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).

7.1 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

8. Hosted Checkout Integration

Introduction

Easypay Online Payment System (OPS) is an electronic payment solution that enables internet users to make financial transactions online. It seamlessly integrates with e-commerce web-sites, shopping carts or mobile apps and allows the owners of these applications to collect payments from their customers. Easypay allows partner businesses to offer their customers a variety of payment methods to pay for their transactions, which include Easypaisa Mobile Accounts, Credit/Debit Card payments.

Hosted Checkout

Hosted Checkout is an extension offered by Mastercard Payment Gateway Services (MPGS) allowing online merchants and e-commerce/webstores to offer customers to pay for their transactions through credit/debit cards directly from the webstore using a payment gateway, without re-directing to the payment gateway's web-page. Easypay OPS offers this feature to merchants or web-stores using Easypay as their payment gateway.

Hosted Checkout allows online merchants to copy a pre-built form in their web-sites' code, to collect payment information from their customers. This form allows customers to input credit card information in the web-site without exposing it to the merchant. Merchant webstores will never handle this information directly. The form will automatically retrieve the information and forward it to Easypay payment gateway for processing..

After integration with the hosted checkout page, merchant can take credit card transaction input (Card Number, CVV, Card Expiry etc.) on their website and send the transaction for processing to Easypay. The credit card details entered on merchant website are secured and cannot be extracted or stored by the web-store's server or database

Security

8.1 Hosted Fields:

Hosted Fields is a feature that allows web developers to outsource the handling of payment card data while developing an ecommerce web-store.

Hosted Fields are input fields that appear on any webpage, but which use JavaScript to ensure that the information entered in these fields is securely handled by an authorized payment gateway such as Easypay's servers rather than merchants' websites/servers where the input fields actually appear. This makes payment processing easier to implement, relieves merchant of having to handle payment details such as credit/debit card information and also prevents unauthorized webstores from storing credit card information.

8.2 Checkout Form:

Checkout Form is the html form which contains Hosted Fields required to input card details. The form works together with the Easypay Hosted Checkout JavaScript file linked with the web-page to secure information retrieval. The form ensures that client side interaction will be between customer browser and OPS server. Merchant and web hosting providers will not have any control.

8.3 Hash Parameter Generation

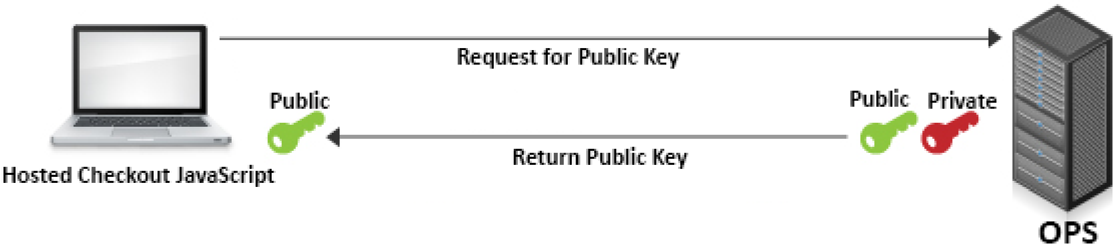
The checkout form also generates a hash string from the user's transaction details and sends it to OPS as an additional parameter. This hash is used by Easypay to identify whether the request has been sent from an authorized merchant webstore.

8.4 Encrypt Card Holder Information

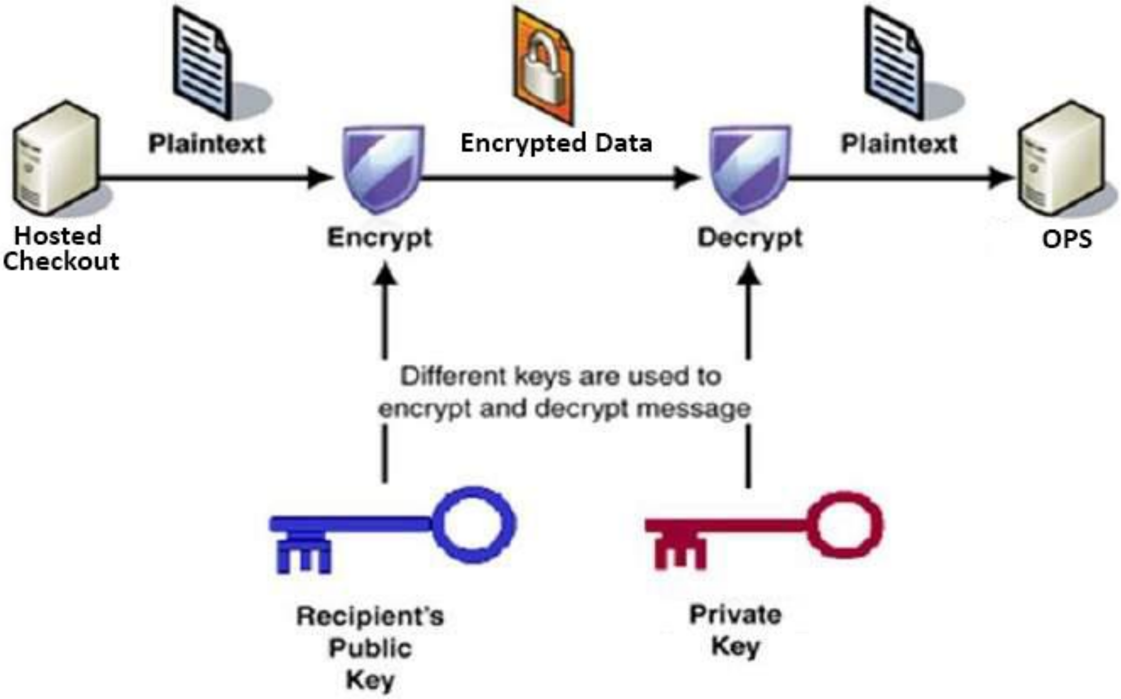
According to PCI DSS regulations, card holders' data must be protected from merchant applications that are not PCI DSS compliant.

The card details in the hosted fields is extracted from customers' browsers by the Hosted Checkout form and is transmitted to OPS Server for payment processing in an encrypted form. The encryption prevents any merchant interference and also protects sensitive information from leaking over the network layer.

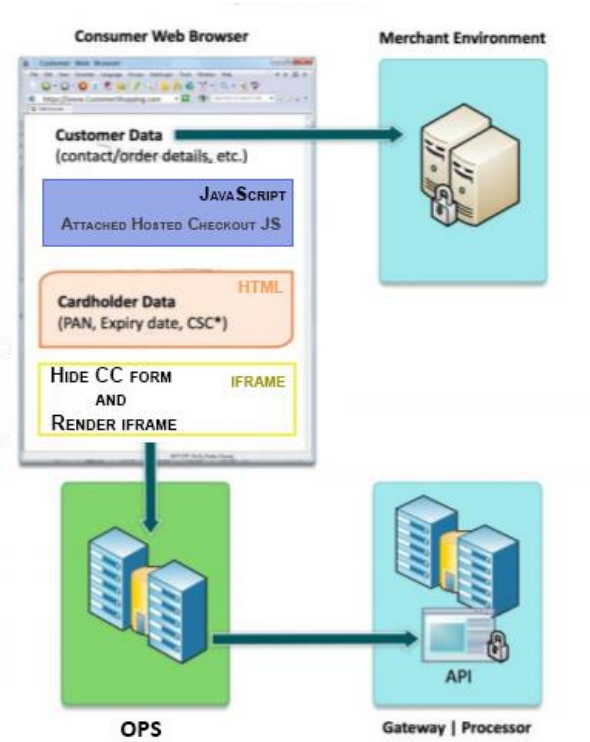
OPS uses the process of Public Key Encryption to secure credit card and card holder information. When users submit their card details via pay button in an e-commerce website, then the Hosted Checkout JavaScript requests the Easypay OPS server for a Public key. This key will be used for transaction information encryption. OPS will encrypt the credit card information using this key and send the encrypted string to OPS for further processing.



After receiving an encrypted request from hosted checkout form, OPS will decrypt the request using its' private key. OPS stores its' private keys in AES encrypted form. These private keys are OPS' own secret and are not shared with anyone or any third party system.



How it works



The checkout page of a web-store, using Hosted Checkout form can be classified into four sections based on their behavior when opened in a customer's browser.

The customer details section contains input fields that collect customer details such as first and last name, address etc. The information from this section is embedded and controlled by the merchants' own web-sites and can be accessed by their web-servers.

The Hosted Checkout URL is a link to the Easypay JavaScript file that must be linked with the merchants' website. This file is used by the Hosted Checkout form in the merchant web-site to secure payment details.

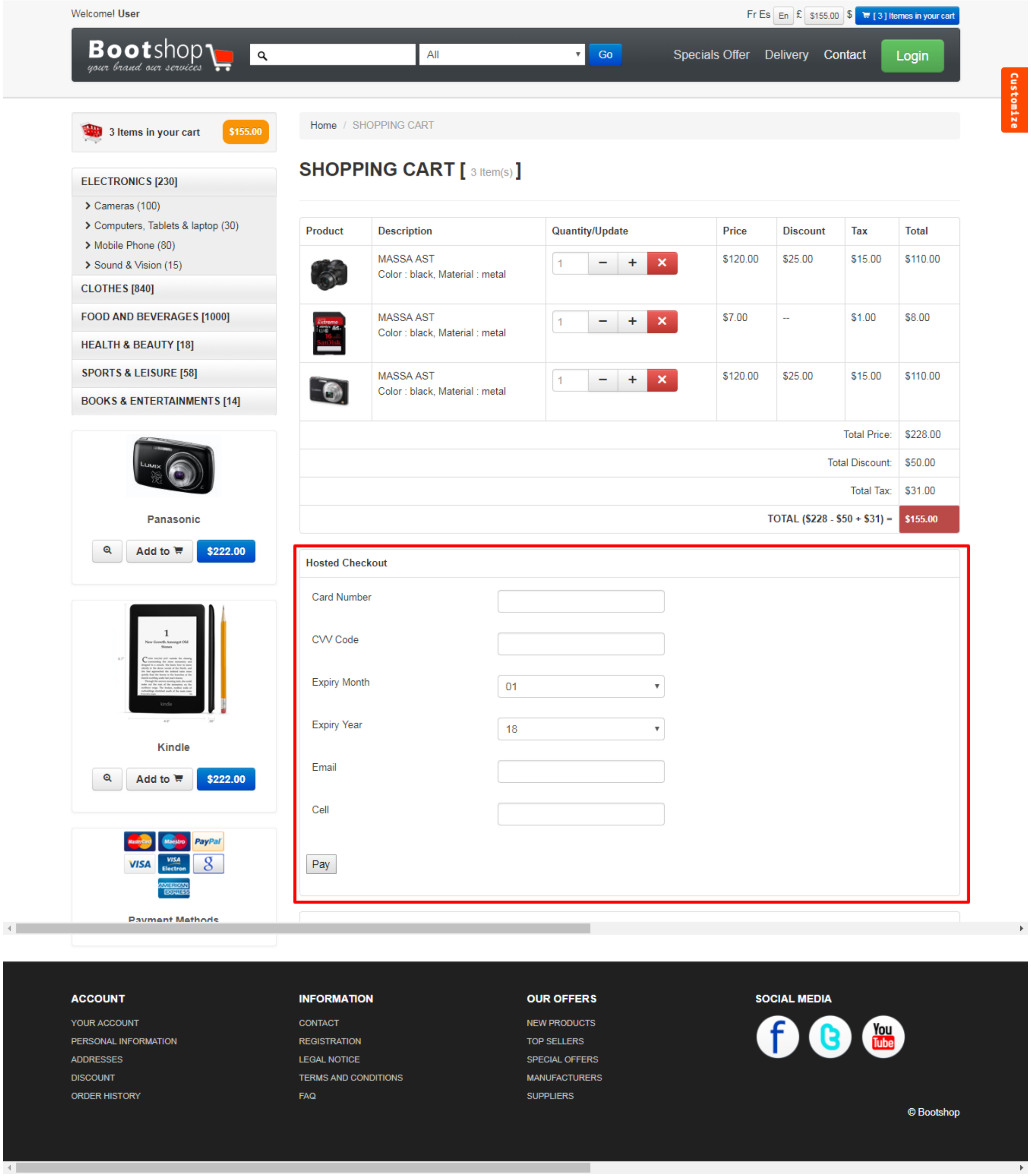
The Hosted Checkout form contains the input fields that collect credit card information i.e. Card number, CVV and Expiry date etc. This form uses the JavaScript files mentioned above, to ensure that the information entered in the hosted fields is only held by the payment gateway and cannot be accessed by merchants' websites.

The payment processor iframe comes with the Hosted Checkout form and is used to display the payment processor's page, within the merchant web-site. The web-page displayed in this iframe is controlled by OPS. OPS in turn is connected with the Mastercard and displays the resulting HTML from Mastercard in this iframe.

8.5 Hosted Checkout User Journey

Step1

The Hosted Checkout form appears on an online store and asks for payment information input from user.



Welcome! User

Fr Es En £ $155.00 $ [3] Itemes in your cart

3 Items in your cart $155.00

ELECTRONICS [230]
- Cameras (100)
- Computers, Tablets & laptop (30)
- Mobile Phone (80)
- Sound & Vision (15)

CLOTHES [840]

FOOD AND BEVERAGES [1000]

HEALTH & BEAUTY [18]

SPORTS & LEISURE [58]

BOOKS & ENTERTAINMENTS [14]

Panasonic — Add to — $222.00

Kindle — Add to — $222.00

Payment Methods

Home / SHOPPING CART

SHOPPING CART [3 Item(s)]

Product	Description	Quantity/Update	Price	Discount	Tax	Total
	MASSA AST Color : black, Material : metal	1	$120.00	$25.00	$15.00	$110.00
	MASSA AST Color : black, Material : metal	1	$7.00	--	$1.00	$8.00
	MASSA AST Color : black, Material : metal	1	$120.00	$25.00	$15.00	$110.00
					Total Price:	$228.00
					Total Discount:	$50.00
					Total Tax:	$31.00
					TOTAL ($228 - $50 + $31) =	$155.00

Hosted Checkout

Card Number	
CVV Code	
Expiry Month	01
Expiry Year	18
Email	
Cell	

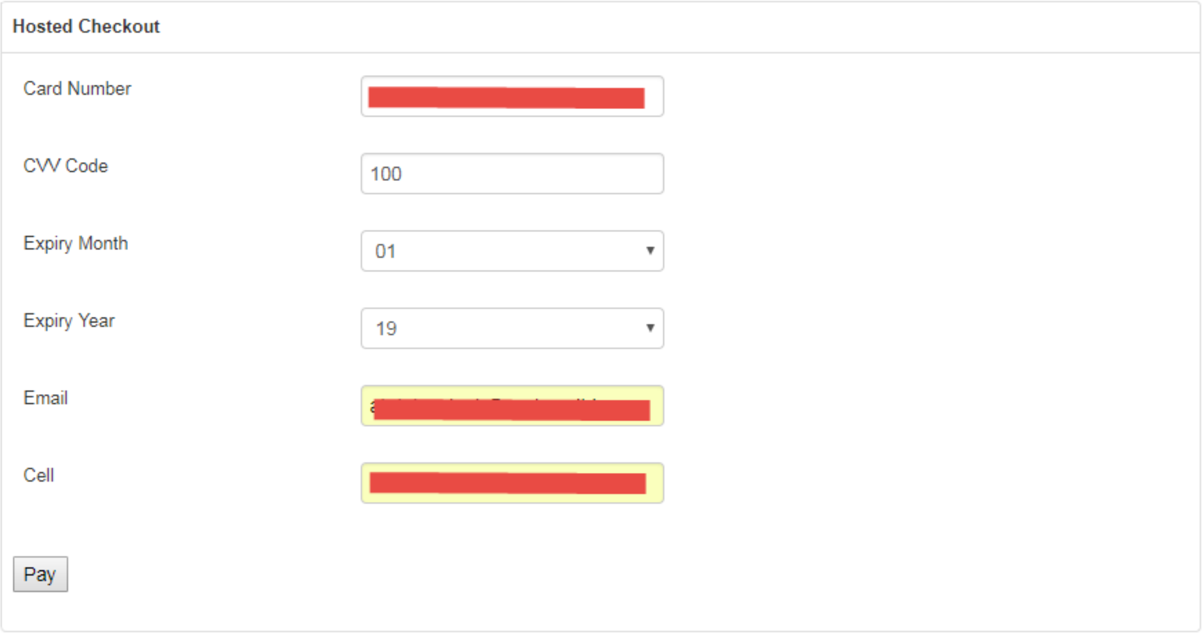
Pay

ACCOUNT	INFORMATION	OUR OFFERS	SOCIAL MEDIA
YOUR ACCOUNT	CONTACT	NEW PRODUCTS	
PERSONAL INFORMATION	REGISTRATION	TOP SELLERS	
ADDRESSES	LEGAL NOTICE	SPECIAL OFFERS	
DISCOUNT	TERMS AND CONDITIONS	MANUFACTURERS	
ORDER HISTORY	FAQ	SUPPLIERS	

© Bootshop

Step2

User enters his/her card details in the Hosted Checkout form and clicks on pay button.

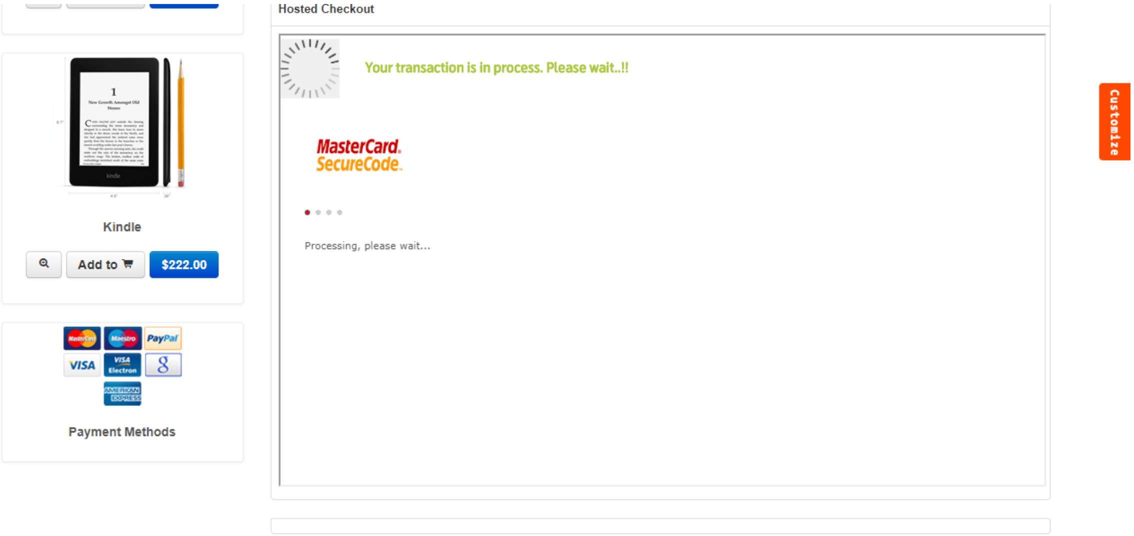


Hosted Checkout

Card Number	[redacted]
CVV Code	100
Expiry Month	01 ▾
Expiry Year	19 ▾
Email	[redacted]
Cell	[redacted]

Step3

The card details form will be hidden and an iframe will appear in its place showing the payment facilitator's processing page.



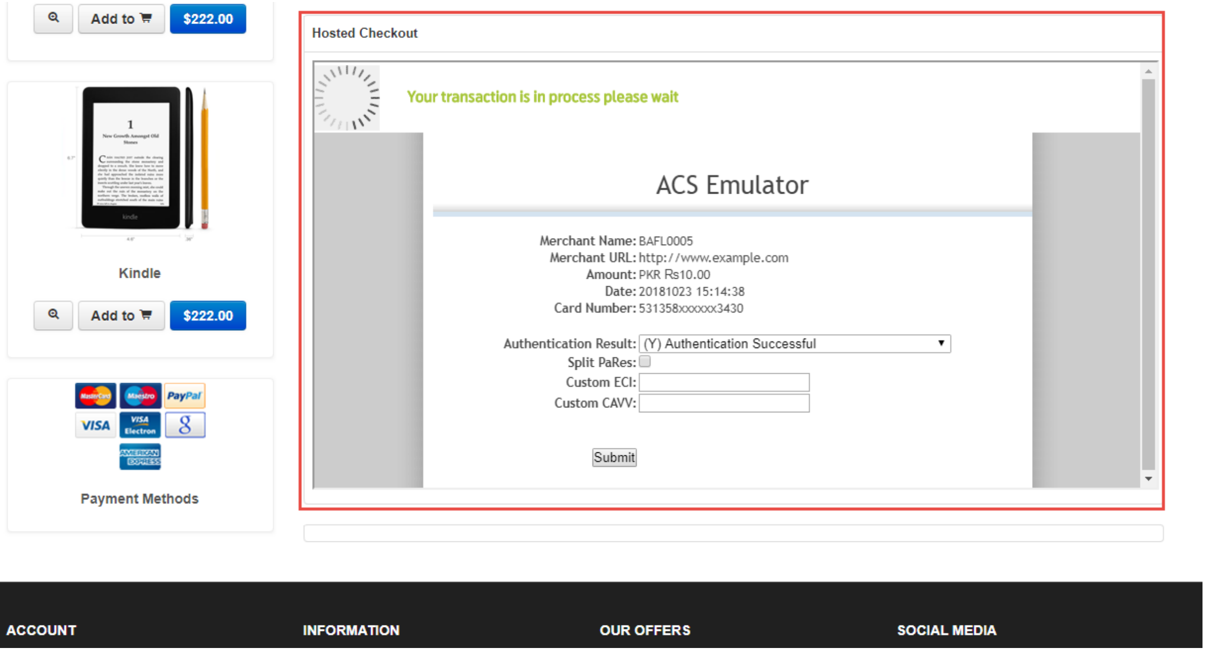
Hosted Checkout

Your transaction is in process. Please wait..!!

Processing, please wait...

Step4 (Conditional)

If the customer’s card is 3d Secured, then the 3DS enrollment check page appears within the iframe.



Hosted Checkout

Your transaction is in process please wait

ACS Emulator

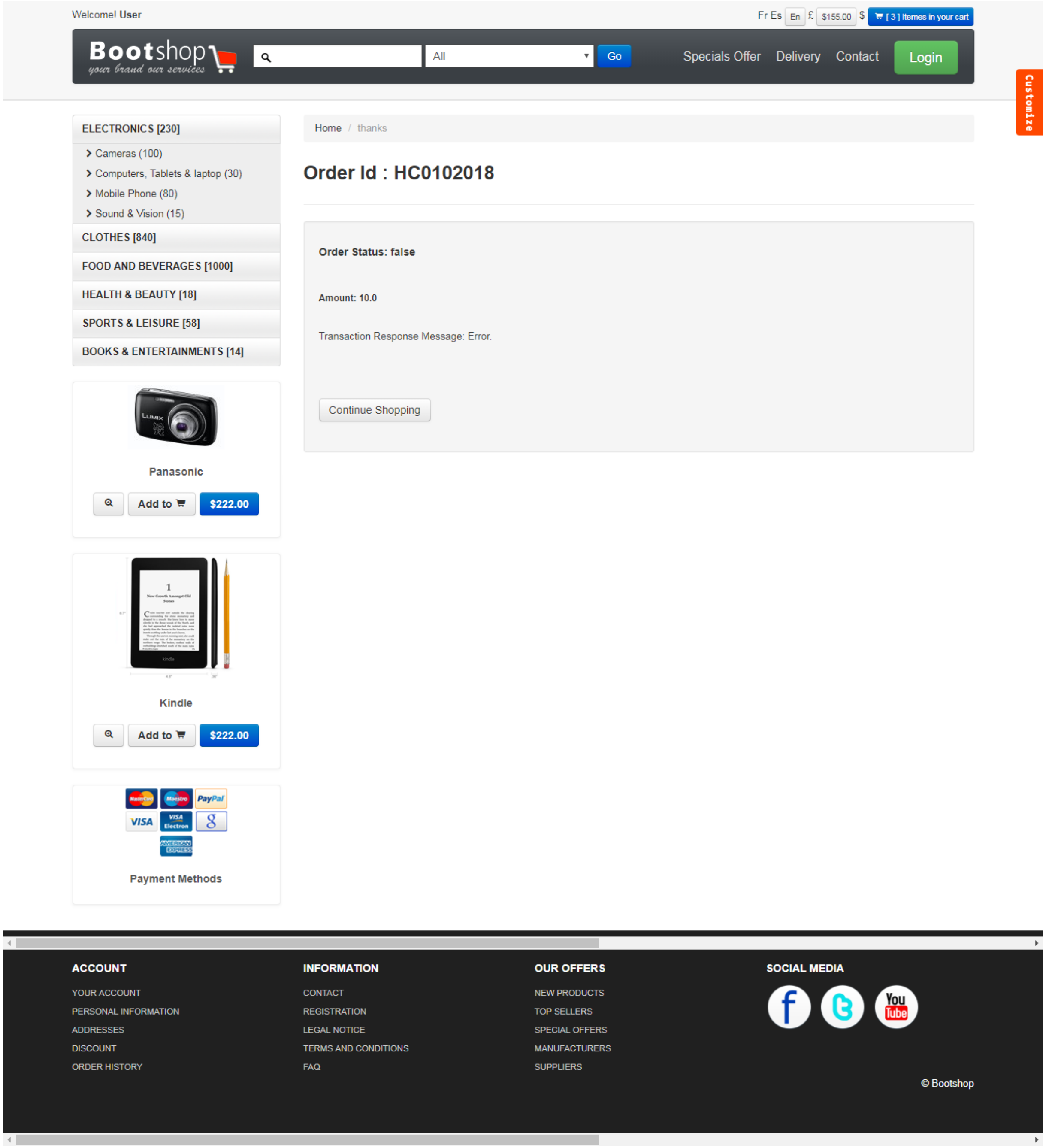
Merchant Name: BAFL0005
Merchant URL: http://www.example.com
Amount: PKR Rs10.00
Date: 20181023 15:14:38
Card Number: 531358xxxxxx3430

Authentication Result: (Y) Authentication Successful ▼
Split PaRes: □
Custom ECI:
Custom CAVV:

Submit

Step5

After all processing, it will automatically redirect to the merchants’ configured thank you page with a set of parameters as shown below.



Order Id : HC0102018

Order Status: false

Amount: 10.0

Transaction Response Message: Error.

8.6 How to integrate

The following sequence of steps needs to be performed to integrate Easypay Hosted Checkout with a merchant website.

Step1

Include the Hosted Checkout form in the order checkout page of the web-store. The following code snippet can be used within the html code body.

```
<div id="hosted-checkout-container">
  <div id="opsMessagesSpan" class="error"></div>
  <iframe id="easypay-iframe" name="easypay-iframe" style="display:none;" width="100%" height="430px"></iframe>
  <form id="hosted-checkout-form">
    <table class="table table-borderless">
      <input type="hidden" id="storeId" value="<?php echo $storeId; ?>">
      <input type="hidden" id="orderRefNumber" value="<?php echo $orderRefNum; ?>">
      <input type="hidden" id="amount" value="<?php echo $amount; ?>">
      <input type="hidden" id="bankIdentificationNumber" value="<?php echo $bankIdentificationNumber; ?>">
      <input type="hidden" id="postBackURL" value="<?php echo $postBackURL; ?>">
      <tr>
        <td>Card Number</td>
        <td>
          <input type="text" id="ccNumber" required/>
        </td>
      </tr>
      <tr>
        <td>CVV Code</td>
        <td>
          <input type="text" id="cvv" required/>
        </td>
      </tr>
      <tr>
        <td>Expiry Month</td>
        <td>
          <select id="expMonth">
            <option value="01">01</option>
            <option value="02">02</option>
            <option value="03">03</option>
            <option value="04">04</option>
            <option value="05">05</option>
            <option value="06">06</option>
            <option value="07">07</option>
            <option value="08">08</option>
            <option value="09">09</option>
            <option value="10">10</option>
            <option value="11">11</option>
            <option value="12">12</option>
          </select>
        </td>
      </tr>
      <tr>
        <td>Expiry Year</td>
        <td>
          <select id="expYear">
            <option value="18">18</option>
            <option value="19">19</option>
            <option value="20">20</option>
            <option value="21">21</option>
            <option value="22">22</option>
            <option value="23">23</option>
          </select>
        </td>
      </tr>
      <tr>
        <td>Email</td>
        <td>
          <input type="text" id="email" value="" required/>
        </td>
      </tr>
      <tr>
        <td>Cell</td>
        <td>
          <input type="text" id="mobileNum" value="" required/>
        </td>
      </tr>
    </table>
    <div id="post">
      <input type="submit" value="Pay" /><span id="validation-message"></span> </div>
  </form>
</div>
```

The following input fields come pre-built in the Checkout form:

- Card Number
- CVV Code
- Expiry Month
- Expiry Year
- Email
- Cell
- Pay

The element name and “id” attribute of every element/tag within the form must be kept the same as in the code snippet above. These IDs are used by the Hosted Checkout JavaScript file to identify the input fields.
Styling of elements within the form can be customized as needed.

Step2

Include links to the following files towards the end of the body in script tags with the link in the “src” attributes and “text/javascript” in the “type” attribute.

1- http://easypay.easypaisa.com.pk/hosted-checkout/js/jquery-1.11.3.js
2- http://easypay.easypaisa.com.pk/hosted-checkout/js/hosted-checkout-4.1.1.js

```
<script src="http://easypay.easypaisa.com.pk/hosted-checkout/js/jquery-1.11.3.js" type="text/javascript"></script>
<script src="http:// easypay.easypaisa.com.pk /hosted-checkout/js/hosted-checkout-4.1.1.js" type="text/javascript"></script>
```

Step3

Save and close the file.

Step4

Your web-site should now be integrated with the Hosted Checkout and should now successfully communicate with Easypay OPS.

8.7 Pre-requisites

In order to use Easypay Hosted Checkout, an active online merchant account is required for your store in Easypay OPS with the following configurations:

- ✓ CC Payment method enabled
- ✓ Hosted Checkout enabled
- ✓ Valid merchant credentials
- ✓ A valid hash-key for your account
- ✓ Hosted Checkout form and JavaScript integrated successfully

"<https:// easypay.easypaisa.com.pk /hosted-checkout/js/hosted-checkout-4.1.1.js>"
"https://easypay.easypaisa.com.pk/hosted-checkout/js/jquery-1.11.3.js"