PayFast

API Implementation Guide

Version 2.2

Contents

Author	Designation
Safdar	Service Delivery
Muhammad Arfeen	Web Architect

Reviewer	Designation
Yousuf Aziz	Head of department

Revision History

Date	Revision No	Revised By	Revision Section	Incorporated Changes
26th September 2018	1.0	S Yousuf Aziz	API End Points	3.5 and 3.9 added
23rd November 2018	1.1	S Yousuf Aziz	API responses	3.3,3.4,3.5,3.7,3.8,3.9,3.14 and 3.15
3rd December 2018	1.2	S Yousuf Aziz	API Parameters	reserved_1, reserved_2, reserved_3 added for future requirement and added security related param for financial API only.
7th February 2019	1.3	S Yousuf Aziz	Error Codes And API responses	Error codes update 3.7 & 3.9
25th June 2019	1.4	S Yousuf Aziz	API's Parameters and Responses	3.3,3.4,3.5,3.6,3.10,3.11,3.14,3.15 & 3.16
20th Sep 2019	1.5	M. Arfeen/ M. Tariq	Additional parameters defined for 3DS API Req/Res	3.10 & 3.11
14th Nov 2019	1.6	M. Arfeen	API description	3.11, 3.12, 3.13, 3.14, 3.15, 3.16
20th Jan 2020	1.7	M. Arfeen	New params: tran_code & tran_narration	3.11, 3.4, 3.6
27th Aug 2020	1.8	M. Arfeen	**cvv** and **customer_ip** parameter requirement in APIs (purchase)	3.5, 3.6 for cvv 3.4, 3.6, 3.11 for customer_ip
1st Sept 2020	1.9	Danyal	Added is non 3ds response	3.3,3.5.3.10 and 3.14
28th Sept 2020	2.0	Danyal	Amended wallet parameters as	3.3,3.10,3.11,3.14

			well as added easypaisa note.	
01st Oct 2020	2.1	Danyal	Amended phone number format along with	3.3,3.4,3.5,3.6,3.10,3.11,3.14
3rd Mar 2021	2.2	Danyal	Adding store_id and currency_code as optional parameters in some API endpoints	3.3,3.4,3,5,3.6,3.10,3.11
22nd April 2021	2.3	Danyal	Adding bill_number and tran_type parameters in some API end points	3.3,3.4,3,5,3.6, 3.10,3.11
14th June 2021	2.4	Danyal	Adding capture and void apis for MIGS transaction	3.17,3.18

1. Preface

1.1 About

This document contains detailed explanation about how to integrate with PAYFAST API has based transactions functionality. This document also contains the details for online transactions.

1.2 Intended Audience

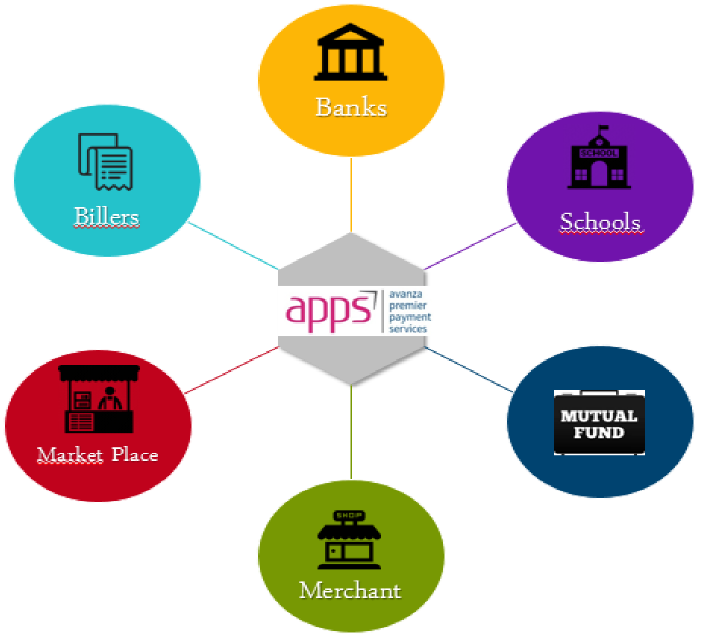
This document is for merchants, acquirers and developers who want to integrate with PAYFAST to perform an API's based Transactions.

1.3 Integration scope

The merchant will implement all ecommerce functionality. PAYFAST' service (***Payfast***) will be used only for payment processing.

2. PAYFAST – Go Cashless!

PAYFAST is focused to bring merchants on ecommerce platforms by offering a complete suite of solutions to all types of merchants from building a website, online store, mobile application, social media marketing to securing payment. PAYFAST will enable merchants to sell the products and services online, and receive their payment safely.

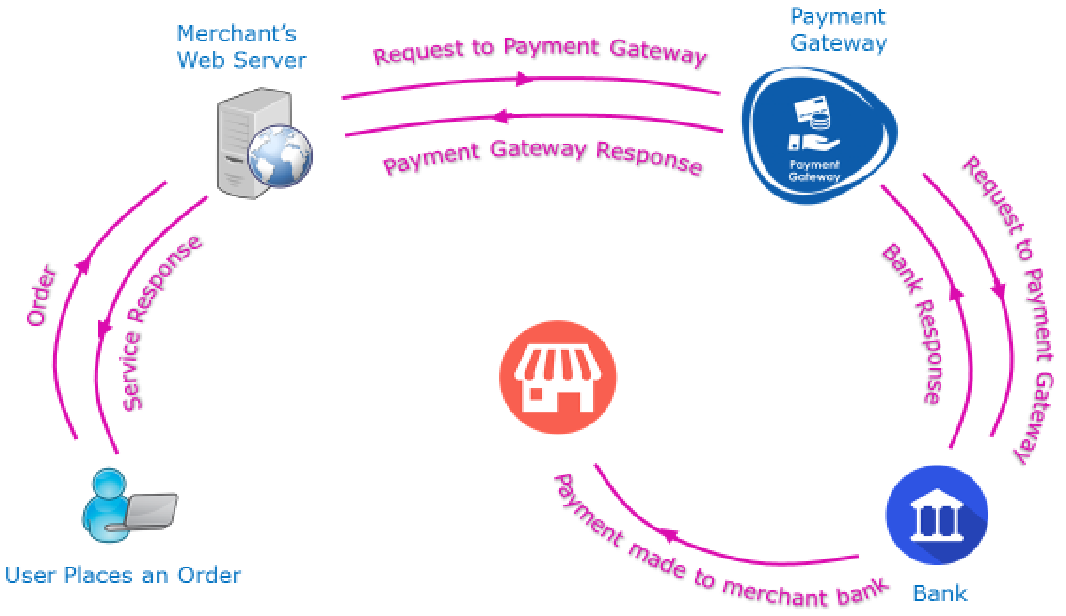


2.1 Key features of PAYFAST:

- Simple, secure and low risk
- Offers easy integration and a broad application suite
- Directly charge from the customer's Bank account, Debit Card and Wallets
- Guaranteed prompt payment to Merchants
- Integrates easily with Mobile and web based application
- Connects to a variety of Merchant organizations including retail stores, governments, etc.

2.2 How *Payfast* Works

PAYFAST has developed a flagship payment gateway solution called "*PayFast*" which is targeted towards merchants, schools and other corporate entities to accept payments digitally from customers by using multiple options such as Debit Cards and Account Numbers. It will be integrated directly with the client's website to allow smooth transaction placements.



2.3 API's based transaction

All retail stores can integrate with ***Payfast*** direct API's for over the counter payments without redirection. After the payment process is completed, the merchant will get an intimation of the transaction status.

2.4 Integration Prerequisites

Merchants will be registered on PAYFAST prior to integration. After merchant sign up for PAYFAST account, following two unique values will be provided to merchant to operate:

- MERCHANT_ID
- SECURED_KEY

These keys are used to get a one-time authentication token, which is used to authenticate payment requests to the "***Payfast***" payment gateway.

2.5 Getting Authentication Token

To initiate the transaction, the merchant needs to generate the Authentication Token from PAYFAST server. For fetching, the token developer needs to call the **Authentication token API.**

2.6 Let's start with Scenario

Scenario 1:

Consider a typical ecommerce transaction where a customer enters card/account details and tries to pay via *PayFast*. In this scenario, a temporary transaction token (refer to section API 3.3) should be fetched (which will last for short period) and then a tokenized transaction API (refer to section API 3.4) will be called to initiate a payment request based on temporary token. In this process, an OTP will be sent to customer mobile number. This token cannot be used for recurring transaction.

Scenario 2:

Consider a scenario where a mobile app wants its customer card information for monthly subscription. This is recurring model, where customer will store its instrument such as card details or account details for later use. It is recommended for merchant to get a temporary transaction (refer to section API 3.3) token against the card details and perform a small amount of transaction (refer to section API 3.4) (which will be refunded) to validate instrument details, based on temporary token. Again, an OTP will be sent to customer and once transaction gets success, merchant will add instruments details and will get the permanent transaction token (refer to section API 3.14) which could be used later on for recurring transaction API (refer to section API 3.6) and no OTP will be sent to customer.

Scenario 3:

Consider a situation where merchant do not want a token against card/account details and want to perform a transaction with the instrument details on the fly. In this case, a simple transaction API (refer to section API 0) will serve the purpose. In this process, merchant would call “validate” (refer to section API 3.10) API to get the OTP for customer and it will be sent with other transaction parameter.

3. API End Points

This section contains the details of all APIs provided by PAYFAST. The merchants, acquirers and/or aggregators could call these APIs. These API’S are based on REST architecture and serve standard HTTP codes for the response payload.

3.1 Authentication Access Token

Following API will provide you the Authentication token, which will be used to call APIs. Merchant_id and Secured_key is mandatory to get the access token. This token will be sent on all the APIs with standard HTTP header ‘Authorization’.

URL	/token
Method	POST
Parameters	merchant_id
	secured_key
	grant_type
	customer_ip (string) End user IP address
	reserved_1 (conditional)
	reserved_2 (conditional)
	reserved_3 (conditional)
	api_version (optional)
Response Type	JSON
Response	{
	"token": "<token>",
	"refresh_token": "<refresh token>",
	"code": "",
	"message": null, "expiry": <no. of seconds> }

3.2 Refreshed Token

Any access token can be refreshed upon expiry. A refresh token is given along with original token.

URL	/refreshtoken
Method	POST
Parameters	grant_type (refresh_token)
	refresh_token (string)
	customer_ip (string) End user IP address
	reserved_1 (conditional)
	reserved_2 (conditional)
	reserved_3 (conditional)
	api_version (optional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON
Response	{
	"token": "<token>",
	"refresh_token": "<refresh token>",
	"code": "",
	"message": null, "expiry": <no. of seconds>
	}

3.3 Get transaction token

Get a temporary payment request token, against card, wallet or account details, which can be used to send a tokenized transaction.

URL	/transaction/token
Method	POST
Common parameters	merchant_user_id (String) user_mobile_number (String) format (0345XXXXXX) instrument_alias (String) -- (optional) account_type (String) customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional) currency_code (string)---(optional) tran_type(string)—(optional) bill_number(string)—(optional) merCatCode (string)---(optional) for only Acquirer store_id (string) – (optional) subMerId (string)---(optional) for only Acquirer subMerName (string)---(optional) for only Acquirer subMerAbbr (string)---(optional) for only Acquirer
Security related Parameter	secured_hash (conditional)
Card Payment Parameters: (UPI,Master,VISA and Paypak Cards)	card_number (String) E.g without spaces/dash expiry_month (String) E.g 09 for September expiry_year (String) E.g 19 for 2019 cvv (String) basket_id (String) order_date (String) txnamt (String) data_3ds_pagemode (SIMPLE/CUSTOMIZED) (required-3DS Only)

	data_3ds_callback_url (required-3DS Only) (see point 6 for 3DS verification process)
Bank Account Payment Parameters:	bank_code (String) account_number (String) account_title (String) -- (optional) cnic_number (String)
Wallet Payment Parameters	bank_code (String) account_number (String) format (0345XXXXXX) account_title (String) -- (optional) cnic_number (String) – for Upaisa only customer_email_address (String) – for easypaisa
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	{
	"status_code": "<status_code>",
	"status_msg": "<status_msg>",
	"instrument_alias": "**********",
	"instrument_token": "********",
	"transaction_id": "<transaction id>",
	“otp_required”: true/false,
	“eci”: true/false,
	"data_3ds_acsurl": <3DSVerificationURL>,
	"data_3ds_pareq": <3DSRequestValue>,
	"data_3ds_html": <3DSRedirectionHTMLPage>,
	"data_3ds_secureid": <3DSecureID>,
	
	"data_3ds_gatewayrecommendation":<3DSRecommendation>,
	“isNon3DsCard”: true/false
	}

Note: Please note that transaction type transaction type (tran_type) parameter is being used for transaction settlement purpose

3.4 Initiate tokenized payment request

Initiate a payment request based on “temporary” transaction token.

URL	/transaction/tokenized
Method	POST
Parameters	instrument_token (String) merchant_user_id (String) user_mobile_number (String) format (0345XXXXXX) basket_id (String) order_date (String) txndesc (String) txnamt (String) otp (string) (conditional) not required when issuer managed OTP transaction_id (string) received from 3.3 API response eci (Boolean) (conditional) received from 3.3 API response) customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional)

	currency_code (string)---(optional) bill_number(string)—(optional) merCatCode (string)---(optional) for only Acquirer store_id (string) – (optional) tran_type(string)—(optional) subMerId (string)---(optional) for only Acquirer subMerName (string)---(optional) for only Acquirer subMerAbbr (string)---(optional) for only Acquirer data_3ds_secureid (String) (required-3DS Only) data_3ds_pares (String) (required-3DS Only) (see point 6 for 3DS verification process) tran_code (String) (optional) tran_narration (String) (optional) (see point 7 for *tran_code* and *tran_narration* explanation)
Security related Parameter	secured_hash (conditional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	{
	"status_code": "",
	"status_msg": "",
	"rdv_message_key": "",
	"basket_id": "",
	"transaction_id": "<transaction id>",
	"code": ""
	}

Note: Please note that transaction type transaction type (tran_type) parameter is being used for transaction settlement purpose

3.5 Getting OTP for recurring transaction

This API will generate an OTP and send to customer, which could be used for sending recurring transaction.

URL	/transaction/recurring/otp
Method	POST
Parameters	instrument_token (String) merchant_user_id (String) user_mobile_number (String) format (0345XXXXXX) basket_id (String) order_date (String) txnamt (String) cvv (for “visa/master card” based transaction) (string) customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional) currency_code (string)---(optional) store_id (string) – (optional) tran_type(string)—(optional) bill_number(string)—(optional) merCatCode (string)---(optional) for only Acquirer subMerId (string)---(optional) for only Acquirer subMerName (string)---(optional) for only Acquirer

	subMerAbbr (string)---(optional) for only Acquirer data_3ds_pagemode (SIMPLE/CUSTOMIZED) (required-3DS Only) data_3ds_callback_url (required-3DS Only) (see point 6 for 3DS verification process)
Security related Parameter	secured_hash (conditional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	{
	"status_code": "<status_code>",
	"status_msg": “<status_msg>",
	"rdv_message_key": "<rdv_message_key>",
	"basket_id": “<basket_id>”,
	"transaction_id": "<transaction id>",
	"code": "null"
	"eci": null,
	"data_3ds_acsurl": “< data_3ds_acsurl>”,
	"data_3ds_pareq": “<data_3ds_pareq>”,
	"data_3ds_html": “<data_3ds_html>”,
	"data_3ds_secureid": “<data_3ds_secureid>”,
	"data_3ds_gatewayrecommendation": “<data_3ds_gatewayrecommendation >”,
	"isNon3DsCard": true/false
	}

Note: Please note that transaction type transaction type (tran_type) parameter is being used for transaction settlement purpose

3.6 Initiate recurring payment request

Initiate payment request based on permanent token. This API will serve recurring payment requests. (Currently this API is not available due to some bank/scheme level constrains)

URL	/transaction/recurring
Method	POST
Parameters	instrument_token (String) merchant_user_id (String) user_mobile_number (String) format (0345XXXXXX) basket_id (String) order_date (String) txndesc (String) txnamt (String) otp (if required (only for UPI, Account & Wallet) (string) cvv (for “visa/master card” based transaction) (string) customer_ip (string) End user IP address eci (Boolean) (conditional) transaction_id (string) received from 3.5 API response reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional) currency_code (string)---(optional) bill_number(string)—(optional) store_id (string) – (optional) tran_type(string)—(optional) merCatCode (string)---(optional) for only Acquirer

	subMerId (string)---(optional) for only Acquirer subMerName (string)---(optional) for only Acquirer subMerAbbr (string)---(optional) for only Acquirer data_3ds_secureid (String) (required-3DS Only) data_3ds_pares (String) (required-3DS Only) **(see point 6 for 3DS verification process)** tran_code (String) (optional) tran_narration (String) (optional) **(see point 7 for *tran_code* and *tran_narration* explanation)**
Security related Parameter	secured_hash (conditional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	{
	"status_code": "",
	"status_msg": "",
	"rdv_message_key": "",
	"basket_id": "",
	"transaction_id": "<transaction id>",
	"code": ""
	}

Note: Please note that transaction type transaction type (tran_type) parameter is being used for transaction settlement purpose

3.7 Issuer/Banks list

This API will provide the available list of issuer/bank

URL	/list/banks
Method	GET
Parameters	customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	"banks": [
	{
	"bank_code": "xxxxxxxx",
	"name": " xxxx",
	"is_slab": true/false,
	"allow_non_islamic": true/false
	}

	], "code": null,
	"message": null
	}

3.8 Getting Payment type (account type/instrument) according to selected bank

This API will provide the Payment type (or account type, e.g. Account, Wallet or Debit Card) based on selected issuer/bank.

URL	/list/instruments
Method	GET
Parameters	bank_code (string) customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON
Response	{
	"bankInstruments": [{ "id": "xxxx", "name": "xxxxxxx" },]
	"code": "00",
	"message": null
	}

3.9 Getting Issuer/Banks list by issuer Instrument ID

This API will fetch the available list of issuer/bank based on instrument id

URL	/list/instrumentbanks
Method	GET
Parameters	instrument_id customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON
Response	
	"banks": [{ "bank_code": "xxxxxxxx", "name": "xxxxxxxxxxxxx" “recurring_allowed”: true/false, “otp_required”: true/false },]
	"code": null,
	"message": null
	}

3.10 Customer Account Validation

This API will allow sending OTP to registered mobile number of the customer of the respective Issuer/Bank.

URL	/customer/validate
Method	POST
Common parameters	basket_id: (string) txnamt: (numeric) order_date (String) customer_mobile_no: (numeric) format (0345XXXXXX) customer_email_address: (string) account_type_id: (numeric) customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional) currency_code (string) --- (optional) store_id (string) --- (optional) tran_type (string) --- (optional) bill_number(string)—(optional) merCatCode (string)---(optional) for only Acquirer subMerId (string)---(optional) for only Acquirer subMerName (string)---(optional) for only Acquirer subMerAbbr (string)---(optional) for only Acquirer
Security related Parameter	secured_hash (conditional)
Card Payment Parameters: (UPI,Master,VISA and Paypak Cards)	card_number (String) E.g without spaces/dash expiry_month (String) E.g 09 for September expiry_year (String) E.g 19 for 2019 cvv (String) data_3ds_pagemode (SIMPLE/CUSTOMIZED) (required-3DS Only) data_3ds_callback_url (required-3DS Only) (see point 6 for 3DS verification process)
Bank Account Payment Parameters:	bank_code (String) account_number (String) account_title (String) -- (optional) cnic_number (String)
Wallet Payment Parameters	bank_code (String) account_number (String) format (0345XXXXXX) account_title (String) -- (optional) cnic_number (String) – for Upaisa only customer_email_address (String) – for easypaisa
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON
Response	{ "code":"", "message":"" "transaction_id": "<transaction id>", “otp_required”: true/false, “eci”: true/false, "data_3ds_acsurl": <3DS Verification URL>, "data_3ds_pareq": <3DS Request Value>, "data_3ds_html": <3DS Redirection HTML Page>, "data_3ds_secureid": <3D Secure ID>, "data_3ds_gatewayrecommendation": <3DS Recommendation> "isNon3DsCard": true/false

	}

Note: Please note that transaction type (tran_type) parameter is being used for transaction settlement purpose

3.11 Initiate Payment Request

This API will allow merchant to initiate payment/transaction request without token

URL	/transaction
Method	POST
Common parameters	basket_id: (string) txnamt: (numeric) order_date (String) customer_mobile_no: (numeric) format (03XXXXXXXX) customer_email_address: (string) account_type_id: (numeric) otp (string) (conditional) not required when issuer managed OTP transaction_id (string) received from 3.10 API response customer_ip (string) End user IP address eci (Boolean) received from 3.10 API response reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional) merCatCode (string)---(optional) for only Acquirer currency_code (string) --- (optional) store_id (string) --- (optional) tran_type (string) --- (optional) bill_number(string)—(optional) subMerId (string)---(optional) for only Acquirer subMerName (string)---(optional) for only Acquirer subMerAbbr (string)---(optional) for only Acquirer tran_code (String) (optional) tran_narration (String) (optional) tran_type (String) --(optional) (see point 7 for *tran_code* and *tran_narration* explanation)
Security related Parameter	secured_hash (conditional)
Card Payment Parameters: (UPI,Master,VISA and Paypak Cards)	card_number (String) E.g without spaces/dash expiry_month (String) E.g 09 for September expiry_year (String) E.g 19 for 2019 cvv (String) data_3ds_secureid (String) (required-3DS Only) data_3ds_pares (String) (required-3DS Only) (see point 6 for 3DS verification process)
Bank Account Payment Parameters:	bank_code (String) account_number (String) account_title (String) -- (optional) cnic_number (String)
Wallet Payment Parameters	bank_code (String) account_number (String) format (0345XXXXXX) account_title (String) -- (optional) cnic_number (String) – for Upaisa only customer_email_address (String) – for easypaisa
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON
Response	{

	"status_code": "", "status_msg": "", "rdv_message_key": "", "basket_id": "", "transaction_id": "<transaction id>", "code": "" }

Note: Please note that transaction type transaction type (tran_type) parameter is being used for transaction settlement purpose

3.12 Get Transaction Details/Status

This API will fetch transaction details with respect to the transaction id or the basket id (provided by merchant).

Details by Transaction ID	
URL	/transaction/<transaction_id>
Method	GET
Parameters	api_version (optional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON
Response	{
	"status_code": "",
	"status_msg": "",
	"rdv_message_key": "",
	"basket_id": "",
	"transaction_id": "<transaction id>",
	"code": ""
	}
Details by Basket ID	
URL	/transaction/basket_id/<basket_id>
Method	GET
Parameters	order_date (String) reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON
Response	{
	"status_code": "",
	"status_msg": "",
	"rdv_message_key": "",

	"basket_id": "",
	"transaction_id": "<transaction id>",
	"code": ""
	}

3.13 Refund Transaction Request

This API will allow merchant to initiate the request for transaction refund in case of any dispute in the transaction.

URL	/transaction/refund/<transaction_id>
Method	POST
Parameters	transaction_id (string) txnamt (Num) refund_reason (String) customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON
Response	{
	"code":"",
	"message":""
	}

3.14 Add Permanent Payment Instrument

This API will allow merchants to add permanent payment instrument on PayFast. A token value will be returned to initiate the transaction for this instrument.

URL	/user/instruments
Method	POST
Common parameters	merchant_user_id (String) user_mobile_number (String) format (0345XXXXXX) instrument_alias (String) -- (optional) customer_email_address (String) – for easypaisa account_type (String) transaction_id (required) (transaction_id received from last any successful transaction) customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional) merCatCode (string)---(optional) for only Acquirer subMerId (string)---(optional) for only Acquirer subMerName (string)---(optional) for only Acquirer subMerAbbr (string)---(optional) for only Acquirer data_3ds_secureid (String) (required-3DS Only) **(see point 6 for 3DS verification process)**
Security related Parameter	secured_hash (conditional)
Card Payment Parameters:	card_number (String)

(UPI,Master,VISA and Paypak Cards)	expiry_month (String) expiry_year (String)
Bank Account Payment Parameters:	bank_code (String) account_number (String) account_title (String) -- (optional) cnic_number (numeric)
Wallet Payment Parameters	bank_code (String) account_number (String) format (0345XXXXXX) account_title (String) -- (optional) cnic_number (String) – for Upaisa only customer_email_address (String) – for easypaisa
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	{
	"status_code": "<status_code>",
	"status_msg": " <status_msg>",
	"instrument_alias": "******",
	"instrument_token": "<instrument_token>",
	"transaction_id": null,
	"otp_required": null,
	"eci": null,
	"data_3ds_acsurl": null,
	"data_3ds_pareq": null,
	"data_3ds_html": null,
	"data_3ds_secureid": null,
	"data_3ds_gatewayrecommendation": null,
	"isNon3DsCard": true/false
	}

3.15 Get List of permanent payment instruments against Username, email, merchant

This API will fetch a list of stored permanent instrument tokens against provided merchant's user's information. (Currently this API is not available due to some bank/scheme level constrains)

URL	/user/instruments
Method	GET
Parameters	merchant_user_id (String) user_mobile_number (String) format (0345XXXXXX) customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional)
Security related Parameter	secured_hash (conditional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	[
	{
	"instrument_token": "<instrument_token>",
	"account_type": "< account_type >",
	"description": "***",
	"instrument_alias": "**********"
	}
	]

3.16 Delete Stored Payment Instrument

This API will allow merchants to delete stored permanent instrument tokens.

URL	/user/instruments
Method	DELETE
Parameters	merchant_user_id (String) user_mobile_number (String) format (0345XXXXXX) instrument_alias (String) instrument_token (String) customer_ip (string) End user IP address reserved_1 (conditional) reserved_2 (conditional) reserved_3 (conditional) api_version (optional)
Security related Parameter	secured_hash (conditional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	{
	"code": Error/Success Code,
	"message": Response Message,
	}

3.17 Capture card payment

This API will allow merchants to capture authorized transactions

URL	/transaction/capture/<transaction_id>
Method	POST
Security related Parameter	secured_hash (conditional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	{
	"code": Error/Success Code,
	"message": Response Message,
	}

3.18 Void card transaction

This API will allow merchants to cancel authorized card transactions

URL	/transaction/void/<transaction_id>
Method	POST
Security related Parameter	secured_hash (conditional)
HTTP HEADER	Authorization: Bearer <token>
Response Type	JSON

Response	{
	"code": Error/Success Code,
	"message": Response Message,
	}

4. Transaction Error Codes & Descriptions

ERROR CODES	ERROR DESCRIPTIONS
00	Processed OK
002	Time Out
001	Pending
97	Dear Customer, you have an insufficient Balance to proceed
106	Dear Customer, Your transaction Limit has been exceeded please contact your bank
3	You have entered an Inactive Account
14	Entered details are Incorrect
55	You have entered an Invalid OTP/PIN
54	Card Expired
13	You have entered an Invalid Amount
126	Dear Customer your provided Account details are Invalid
75	Maximum PIN Retries has been Exceeded
14	Dear Customer, You have entered an In-Active Card number
15	Dear Customer, You have entered an In-Active Card number
42	Dear Customer, You have entered an Invalid CNIC
423	Dear Customer, We are unable to process your request at the moment please try again later
41	Dear Customer, entered details are Mismatched
801	{0} is your PayFast OTP (One Time Password). Please do not share with anyone.
802	OTP could not be sent. Please try again later.
803	OTP has been sent to your email address.

804	OTP has been sent to your mobile number.
805	OTP Verified
806	OTP could not be verified.
807	Too many attempts. Please try again later in few minutes.
808	Passwords do not match
809	Invalid Password
810	Password could not be changed
811	Password changed successfully
812	Request could not be validated. Please try again.
813	Email address already registered
850	OTP not required because issuer manages OTP itself.
851	OTP required for permanent token.
79	Alternate Success response
9000	Rejected by FRMS

5. Hashed Parameters list

API Reference	Parameters to be Hashed	Example Data To Be Hashed
3.3	merchant_user_id	**For Card :-** merchant_user_id+ user_mobile_number+card_number+ expiry_month+expiry_year+cvv **For Account :-** merchant_user_id+ user_mobile_number+account_number+ cnic_number **For Wallet: -** merchant_user_id+ user_mobile_number+account_number+ cnic_number
	user_mobile_number	
Card	card_number expiry_month expiry_year cvv	
Account	account_number cnic_number	
Wallet	account_number cnic_number	
3.4	instrument_token merchant_user_id user_mobile_number txnamt otp	instrument_token + merchant_user_id + user_mobile_number + txnamt + otp
3.5	instrument_token merchant_user_id user_mobile_number	instrument_token + merchant_user_id + user_mobile_number
3.6	instrument_token merchant_user_id user_mobile_number txnamt basket_id otp (if applicable)	instrument_token + merchant_user_id + user_mobile_number + txnamt + basket_id [+ otp]
3.10		
Common	basket_id txnamt	
Card	card_number expiry_month expiry_year	basket_id + txnamt + card_number+ expiry_month+expiry_year+cvv

	cvv	basket_id + txnamt + account_number + cnic_number basket_id + txnamt + account_number + cnic_number
Account	account_number cnic_number	
Wallet	account_number cnic_number	
3.11		
Common	basket_id txnamt otp	
Card	card_number expiry_month expiry_year cvv	basket_id + txnamt + card_number+ expiry_month+expiry_year+cvv+otp basket_id + txnamt + account_number + cnic_number+otp basket_id + txnamt + account_number + cnic_number+otp
Account	account_number cnic_number	
Wallet	account_number cnic_number	
	txnamt	
	otp	
3.14		
	merchant_user_id	merchant_user_id+ user_mobile_number+card_number+ expiry_month+expiry_year+cvv merchant_user_id+ user_mobile_number+account_number+ cnic_number merchant_user_id+ user_mobile_number+account_number+ cnic_number
	user_mobile_number	
Card	card_number expiry_month expiry_year cvv	
Account	account_number cnic_number	
Wallet	account_number cnic_number	
3.15		
	merchant_user_id user_mobile_number	merchant_user_id+ user_mobile_number
3.16		
	merchant_user_id user_mobile_number instrument_token	merchant_user_id + user_mobile_number + instrument_token

Note:

1. *"+" symbol denotes simple string concatenation.*
2. *Hashed data will be received in* **"secured_hash"** *parameter in designated APIs*
3. A key will be provided separately for the hash calculation.

Hash Method: HMAC (Keyed Hash)
Hash Algorithm: sha256
Hash Key: ASCII encoded string

6. 3D Secure Transaction

In case of Visa/Master card transaction, 3DS verification will be performed.

a. APIs 3.10, 3.3, 3.5, will accept an additional parameter "data_3ds_pagemode", it may have two values: CUSTOMIZED / SIMPLE.
   a. In "CUSTOMIZED" case, **data_3ds_acsurl** will be returned in APIs 3.10, 3.3, 3.5, which contains 3DS verification URL where the customer should be redirected to verify the transaction with OTP.
      i. To redirect customer, a form will be posted to "**data_3ds_acsurl**" along with following fields:
         1. PaReq: < data_3ds_pareq> (will be returned in response in APIs 3.10, 3.3, 3.5)
         2. TermUrl: Call Back URL (URL where customer will be returned after 3DS verification)
         3. MD: "" (empty value)
   b. In "SIMPLE" case, "**data_3ds_html**", will contain a pre-formatted html page, which can be used directly to redirect customer to verify OTP. TermUrl will be prepopulated due to **data_3ds_callback_url** parameter in APIs 3.10, 3.3, 3.5.

b. APIs 3.11, 3.4, 3.6, will accept 2 more additional parameters for 3DS verification:
   a. **data_3ds_secureid:** This will be returned in response of APIs 3.10, 3.3, 3.5 (Customer Validation).
   b. **data_3ds_pares:** This value will be returned in call back URL, which is called after 3DS verification, the parameter, is returned 'paRes'.

7. Parameters: tran_code And tran_narration

1. Following is a list of valid transaction codes (tran_code):

0020	PURCHASE
0030	CHEQUE DEPOSIT

2. (tran_narration) should contain information in following pattern (delimited by "|" (pipe) sign):

FROM_BANK_NAME|TO_BANK_NAME|ACCOUT_PAYEE|INSTRUMENT_NUMBER (e.g. Cheque no.)

e.g.:
HBL|Meezan Bank|M. Khan|A000000000010

*Note: For easy paisa transaction, user should enter PIN in case of appearance of USSD, or should allow pending approval transaction in case of mobile app, so that API could fetch proper response against the payment request.