

About:

This file is a technical guide for integration of merchant backend with Please refer to below guidelines for integration.

A.List of Files

1. txn.pug - Merchant Data collection.
2. Index.js - Variable Configuration file , Merchant Data encryption and sending data to airpay ,Response from airpay .

B.Data captured.

	Data	HTML variable name	Data Format	Data Length	Example
	Buyer Email	buyerEmail	email format	6-50	eg.buyer@example.com
	Buyer Phone	buyerPhone	numeric, space, hyphen	8-15	9898989989898
*	Buyer First Name	buyerFirstName	alphanumeric, space	1-50	name 89
*	Buyer Last Name	buyerLastName	alphanumeric, space	1-50	name 89
	Buyer Address	buyerAddress	alphanumeric, space and these special chars ,;#\$%&'()-_	4-50	
	Buyer City	buyerCity	alphanumeric, space	2-50	Dodomo
	Buyer State	buyerState	alphanumeric, space	2-50	Dodomo
	Buyer Country	buyerCountry	alphanumeric, space	2-50	Tanzania
	Buyer Pincode	buyerPinCode	alphanumeric	4-8	4124578
	Payment Mode	chmod	chars		ppc_pg_nb(ppc - prepaid card,pg- payment gateway, nb- netbanking)***
*	Amount	amount	number with two decimals	1-6.2	999999.99
*	Order ID	orderid	alphanumeric	1-20	
*	Private Key	privatekey	alphanumeric	10-200	As per calculated in the example.
*	Merchant ID	mercid	numeric	1-12	
*	Checksum	checksum	alphanumeric	10-200	As per calculated in the example.
*	Currency	currency	numeric	3	834**
*	ISO Currency	isocurrency	chars	3	TZS**
*	Custom variable	customvar	alphanumeric,space,=	1-100	eg.airpay Payment
*	Airpay Version	arpyVer	alphanumeric	1	3***
	Sub Type	txnsbtype	numeric	1-100	eg.1000

* represents mandatory.

** Hardcode these example values

***chmod variable contains Payment Modes available for user. for e.g. If you want to show only Credit Card/Debit Card, then value of the chmod variable will be "pg". If you want Netbanking and Prepaid card then value of the chmod variable will be "nb_ppc".If you want to show all payment options activated for you at airpay, then leave this variable blank.

C.Details.

1. transaction.pug

This file has example of data collection from merchant. It collects following data.

Buyer Email, Buyer Phone, Buyer First Name, Buyer Last Name, Buyer Address, Buyer City, Buyer State, Buyer Country, Buyer Pincode, Amount, Order ID.

orderid variable is kept here, so that you can test the kit conveniently. orderid should be unique reference number generated and send by your system to airpay. So you may generate it in sendtoairpay.php

2. index.js

This file contains variables username, password, secret, mercid which you can get from settings page of your airpay merchant account. This variables are used in sendtoairpay.pug and response.pug files. privatekey and checksum are generated using these variables.

In this file, a token generation logic has been implemented, utilizing request data and a token URL. The generated token is subsequently decrypted and appended to the URL.

This file also provides you example of how process data received from transaction.pug and send it to airpay payment solutions.

We are also collecting two other HTML variables.

- currency: 834
- isocurrency: TZS

This file also states how to process response from airpay. Airpay will send you following list of variables.

- TRANSACTIONID = orderid you have send to airpay system
- APTRANSACTIONID = airpay transaction reference number
- AMOUNT = transaction amount
- TRANSACTIONSTATUS = successfull = 200
- MESSAGE = Response message received from payment gateway.
- ap_SecureHash = Secure hash generated by airpay
- CUSTOMVAR = customvar value received from payment gateway.
- All fields are mandatory except MESSAGE.
- In the file we have done different validations and mentioned what necessary action needs to be taken. Please go through php comments mentioned in this file.
- Secure hash is calculated using crc-32. More details are available on <https://www.npmjs.com/package/crc-32> mercid = Merchant Id, username = username available on on settings page of your airpay merchant account.

D. NPM Packages

express-validator 6.14.0

express-validator is a set of express.js middlewares that wraps validator.js validator and sanitizer functions.

Installation

Install it using npm:

```
npm install -save express-validator
```