

**About:**

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

**A. List of Files**

1. Index.cshtml - Merchant Data collection.
2. Config.cs - Variable Configuration file.
3. Functions.cs - checksum, encryption and decryption functions.
4. IndexController.cs - Merchant Data encryption and server side validation and accept response from server.
5. Sendtoairpay.cshtml - sending data to airpay.
6. Successairpay.cshtml - Response page for success transaction.
7. Failureairpay.cshtml - Response page for failed or cancelled transaction.
8. Validate.cshtml - Landing page for error.
9. Crc32.cs - Contains code that refer for secure hash matching.
10. style.css and images (airpay-text-wh.svg and covering.png) - To add style.

Note: Also include .cs file for all .cshtml files to define Pagemodel for corresponding .cshtml files. If the dotnet version does not have option to add controller put following code inside Program.cs file.

```
builder.Services.AddControllers();
app.MapControllerRoute();
```

**B. Data captured.**

	Data	HTML variable name	Data Format	Data Length	Example
	Buyer Email	buyerEmail	email format	6-50	<a href="mailto:eg.buyer@example.com">eg.buyer@example.com</a>
	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
*	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
*	Custom variable	customvar	alphanumeric,space,=	1-100	eg.airpay Payment
	Sub Type	txnsbtype	numeric	1-100	eg.1000
*	Airpay Version	arpyVer	numeric	1-100	3

\* 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. Index.cshtml**

This file has example of data collection from merchant. Place this file inside 'Pages' folder. 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.cshtml

**2. Config.cs**

This file contains variables username, password, secret, mercid,clientID,clientSecret,URL,tokenURL which you can get from settings page of your airpay merchant account. These variables are used in IndexController.cs file. privatekey and checksum are generated using these variables. Place this file inside Pages->Shared folder.

### 3. Functions.cs

This file contains encryption,decryption,checksum and sendpostdata functions.

### 4. IndexController.cs

This file provides you example of how process data received from Index.cshtml and also states how to process response from airpay and send data to corresponding page.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. Airpay will send you following list of variables. Place this file inside 'Controllers' folder.

- 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 and CUSTOMVAR .
- Secure hash is calculated using crc32. More details are available on <http://in2.rb.net/crc32>
- mercid = Merchant Id, username = username available on settings page of your airpay merchant account.

### 5. Sendtoairpay.cshtml

This file data received from IndexController.cs file and send it to airpay payment solutions. Place this file inside Pages->Shared folder.

- currency: 834
- isocurrency: TZS

### 6. Successairpay.cshtml

This file display the message from airpay if the transaction is success. Place this file inside Pages->Shared folder.

### 7. Failureairpay.cshtml

This file display the message from airpay if the transaction is fail or cancelled. Place this file inside Pages->Shared folder.

### 8. Validate.cshtml

Example landing page if server side validation failed for transaction.html.erb. Place this file inside 'Pages' folder.

### 9. Crc32.cs -

Contains code that refer for secure hash matching.

### 10. style.css and images -

Place style file inside wwwroot->css folder and image files (airpay-text-wh.svg and coveringimg.png) inside wwwroot->images file.

## Installation

Ensure you have the .NET SDK version 8.0 installed. You can download it from the [.NET download page](#).

## Rendering the client page

1. Open Command Prompt
2. Navigate to Your Project Directory
3. Use this command : dotnet run --launch-profile dotnetkit

