Worldline – IPG

Merchant Integration Kit - Card Capture Approach

Disclaimer

This documentation shall only be used for evaluating the planned services designated herein, and may contain information that is privileged, confidential, Proprietary or otherwise protected from disclosure. As a result, this document or content thereof shall not be disclosed, used or duplicated, in whole or in part, for any purpose other than the Scope of Work assigned by Worldline India Private Limited ("WL India") to your company ("Recipient"). Upon completion of service or termination of service, the Recipient shall return all materials, including, without limiting the generality of the foregoing, all originals, copies, reproductions and summaries of confidential information. Any unauthorized use or disclosure by the directors, officers or employees of the Recipient shall be deemed to be unauthorized use or disclosure by the Recipient and the Recipient shall indemnify and hold harmless the WL India from and against any and all damages, losses, costs and expenses incurred as a result of such breach.

WL India may seek injunctive relief restraining the unauthorized disclosure or use of confidential information in addition to any other legal or equitable remedy otherwise available.

Recipient is not authorized to share this document with any other entity without taking written consent from WL India.

Table of Contents

1 Introduction

1.1 Purpose

This document details out the technical integration approach between WL PG and the Merchants. It also describes the WL PG Client API functionalities that provide quick integration solution for Merchants.

1.2 Prerequisites

The merchant should have the following pre-requites in place before integrating with AWL PG.

1. Merchant needs to be on boarded and must have a valid Merchant ID.
2. Merchants should have a valid Merchant Encryption Key that is generated & sent to Merchants on successful onboarding process.

1.3 API List

WL PG provides below API's to the Merchants as part of Payment Gateway:

1. Transaction API.
2. Transaction Status Enquiry API
3. Transaction Cancellation API
4. Transaction Refund API

2 API Integration

2.1 Transaction API

Transaction API helps merchant to integrate with WL PG for transaction process. This API contains the two services as follows:

1. Generate Payment Request: Using this service, Merchants can generate a Payment Request Message for it to be served as an input parameter during the transaction call. It receives the Payment Request parameters and encrypts them using the Merchant key shared by WL.

   Once the Request Message is generated, Merchant can then redirect the customer to WL PG along with the Request Message.

2. Parse Payment Response: Post Payment process, WL PG will redirect the customer back to Merchant site with a Response Message. Merchant application calls this service to decrypt this Response Message and thereby allowing them to interpret the Payment status passed from WL PG.

2.1.2 Payment Request Process

Request Parameters:

Sr. No.	Fields	Description	Mandatory (M) / Optional (O)	Data Type	Max Length
1	MID	Merchant ID	M	Character	15
2	Order ID	Unique Order Reference sent by merchant	M	Character	30
3	Transaction Amount	Transaction amount in **Paisa** Format	M	Numeric	15
4	Transaction Currency	Currency code of the transaction. Example: INR	M	Character	5
5	Transaction Description	Order Summary Description	M	Character	100
6	Transaction Type	The following are the transaction types: Normal Transaction: **S** Pre-Auth Transaction: **P**	M	Character	2
7	Response URL	Merchant Response URL	M	Character	500

8	Card Number	Card Number	M	Numeric	19
9	Expiry Date	Expiry Date (format: MMYY)	M	Numeric	4
10	CVV	Card CVV	M	Numeric	4
11	Name On Card	Name on Card	M	Character	20
12	Payment Type	**CC**: Credit Card / **DC**: Debit Card	M	Character	2
13	Addl Field 1	Additional field for future use	O	Character	100
14	Addl Field 2	Additional field for future use	O	Character	100
15	Addl Field 3	Additional field for future use	O	Character	100
16	Addl Field 4	Additional field for future use	O	Character	100
17	Addl Field 5	Additional field for future use	O	Character	100
18	Addl Field 6	Additional field for future use	O	Character	100
19	Addl Field 7	Additional field for future use	O	Character	100
20	Addl Field 8	Additional field for future use	O	Character	100

Technical Integration Steps:

1. On click of CheckOut/Pay button, Merchant application will generate a Payment Request message.
2. Merchant application should pass following parameters to generate a Request Message:
   - **a.** Merchant ID: Shared by WL.
   - **b.** Order No: Merchant Unique Reference No. to identify an order.
   - **c.** Transaction Amount: In Paisa format.

d. Transaction Currency.

e. Payment Description.

f. Transaction: Value can be either **"S" or "P"** as described in the table above.

g. Return URL.

h. Encryption Key: Shared by WL.

i. Payment Type (**CC**: Credit Card / **DC**: Debit Card).

j. Card Number.

k. Expiry Date.

l. CVV.

m. Name on Card.

Sample Code to generate the payment request message:

```
//Step 1: include the following namespace
<?php
include 'AWLMEAPI.php';

//Step 2: Assign payment request parameter values to the DTO
$obj = new AWLMEAPI();

$reqMsgDTO = new ReqMsgDTO();

$reqMsgDTO->setOrderId("1000001");
$reqMsgDTO->setMid("AWL000000000001");
$reqMsgDTO->setTrnAmt(200);          //Paisa Format
$reqMsgDTO->setTrnCurrency("INR");
$reqMsgDTO->setMeTransReqType("S");
$reqMsgDTO->setEnckey("4f5390bey3ef1ee3d4a7e77fd42238cb");
$reqMsgDTO->setResponseUrl("http://merchant.com/meTrnPay.aspx");
$reqMsgDTO->setTrnRemarks("Mobile bill paid");
$reqMsgDTO->setCardNumber("5420343024239514");
$reqMsgDTO->setExpiryDate("1220");
$reqMsgDTO->setCvv("123");
$reqMsgDTO->setNameOnCard("Test");
$reqMsgDTO->setPayTypeCode("CC"); //CC or DC

//Optional Fields

$reqMsgDTO->setAddField1("Info1");
$reqMsgDTO->setAddField2("Info 2");
$reqMsgDTO->setAddField3("");
$reqMsgDTO->setAddField4("");
$reqMsgDTO->setAddField5("");
$reqMsgDTO->setAddField6("");
$reqMsgDTO->setAddField7("");
$reqMsgDTO->setAddField8("");
//Step 3: Call API to generate the message

$merchantRequest = "";

$reqMsgDTO = $obj->generateTrnReqMsg($reqMsgDTO);

        if ($reqMsgDTO->getStatusDesc() == "Success"){
                $merchantRequest = $reqMsgDTO->getReqMsg();
}
?>
```

3. Post Payment Request Message generation, redirect the customer to WL PG.

Refer Section 3.2 for configuring Test/Production form action URL

Sample Code:

```
<form id="AWLPGPost"
action=”https://cgt.in.worldline.com/ipg/doMEPayRequestCard” method="POST" >

<input type="hidden"  name="merchantRequest "  value="Generated message" />
<input type="hidden"  name="MID"  value="Merchant ID"/>

<input type=”submit” name=”checkout” value=”Checkout”>
</form>
```

2.1.3 Payment Response Process

Response Parameters:

Sr. No.	Fields	Description	Data Type	Max Length
1	Transaction Reference No	Transaction Reference No Provided by AWL PG	Number	16
2	Order ID	Unique Order Reference sent by Merchant	Character	30
3	Amount	Transaction amount in paisa format.	Character	15
4	Status Code	Transaction Status **S**: Success / **F**: Failed	Character	1
5	Status Description	Detailed description of transaction status	Character	100
6	RRN	Bank Reference Number	Character	20
7	Authzcode	Authorisation code sent by bank	Character	6
8	Response code	Response code sent by bank	Character	3
9	Transaction Date Time	Transaction date time (yyyy-MM-dd HH:mm:ss)	Character	20
10	Addl Field 1	Additional field for future use	Character	100
11	Addl Field 2	Additional field for future use	Character	100
12	Addl Field 3	Additional field for future use	Character	100
13	Addl Field 4	Additional field for future use	Character	100
14	Addl Field 5	Additional field for future use	Character	100
15	Addl Field 6	Additional field for future use	Character	100
16	Addl Field 7	Additional field for future use	Character	100

17	Addl Field 8	Additional field for future use	Character	100

Technical Integration Steps:

1. On receipt of Transaction Response from WL PG, Merchant shall call the API to parse the Response Message and interpret the Status received in the Response.

   Sample Code to receive the transaction response message

```
<?php

include 'AWLMEAPI.php';

$obj = new AWLMEAPI();
$resMsgDTO = new ResMsgDTO();
$reqMsgDTO = new ReqMsgDTO();
$enc_key = "df58d8ce4f9ce8a7865fa7b08e13f2e5";

$responseMerchant = $_REQUEST['merchantResponse'];

$response = $obj->parseTrnResMsg( $responseMerchant , $enc_key );

if ($response->getStatusCode()=="S")
        //Success
else
        //Failed
?>
```

2.2 Transaction Status Enquiry

Transaction Status API facilitates the Merchants to enquire on the status of a transaction:

Request Parameters:

Sr. No.	Fields	Description	Mandatory (M) / Optional(O)	Data Type	Max Length
1	MID	Merchant ID	M	Character	15
2	Order ID	Unique Order Reference sent by Merchant	M	Character	30

3	PG Transaction Reference No	PG Transaction Reference No	O	Number	16

Sample code to get Transaction status:

```
//Step 1: include the following namespace
<?php
include 'AWLMEAPI.php';
$obj = new AWLMEAPI();

$orderId = "1000001";
$mId ="AWL000000000001";
$enc_key="4f5390bey3ef1ee3d4a7e77fd42238cb";
$pgMeTrnRefNo="";
//Step 2: Construct the request DTO with respective Parameter

$resMsgDTO = $obj->getTransactionStatus( $mId , $orderId , $pgMeTrnRefNo , $enc_key);


//Step 4: Retrieve Status:
if ($resMsgDTO->getStatusCode()== "S"){
        //Success
else
        //Failed
}
?>
```

2.3 Cancel Transactions

Transaction Cancellation API facilitates the Merchant to cancel a transaction in case the Merchant was unable to fulfil an Online order (or) any technical error that prevented transaction completion status at Merchant end.

A Transaction can **ONLY** be cancelled, if it has not yet settled.

Request Parameters:

Sr. No.	Fields	Description	Mandatory(M) / Optional(O)	Data Type	Max Length
1	MID	Merchant ID	M	Character	15
2	Order ID	Unique Order Reference sent by Merchant.	M	Character	30

3	PG Transaction Reference No	PG Transaction Reference No.	M	Number	16
4	Addl Field 1	Additional field for future use	O	Character	100
5	Addl Field 2	Additional field for future use	O	Character	100
6	Addl Field 3	Additional field for future use	O	Character	100
7	Addl Field 4	Additional field for future use	O	Character	100
8	Addl Field 5	Additional field for future use	O	Character	100
9	Addl Field 6	Additional field for future use	O	Character	100
10	Addl Field 7	Additional field for future use	O	Character	100
11	Addl Field 8	Additional field for future use	O	Character	100

Sample code to call Transaction cancel Request:

```
//Step 1: include the following namespace
<?php
include 'AWLMEAPI.php';

$obj = new AWLMEAPI();

$reqMsgDTO = new ReqMsgDTO();
$reqMsgDTO->setOrderId("1000001");
$reqMsgDTO->setMid("AWL000000000001");
$reqMsgDTO->setEnckey("4f5390bey3ef1ee3d4a7e77fd42238cb");
$reqMsgDTO->setPgMeTrnRefNo(100065);

//Step 3: Call API to get the Cancel API
$resMsgDTO = $obj->cancelTransaction($reqMsgDTO);


//Step 4: Retrieve Status:
if ($resMsgDTO->getStatusCode()==“S”)
        //Success
else
        //Failed
?>
```

Response Parameters:

Sr. No.	Fields	Description	Mandatory(M) / Optional(O)	Data Type	Max Length
1	Order ID	Unique Order Reference sent by Merchant	M	Character	30
2	PG Transaction Reference No	PG Transaction Reference No	M	Number	16
3	Status Code	Cancellation Request status(**S**-Success/**F**-Fail)	M	Character	1
4	Status Description	Cancellation Request status description	M	Character	100
5	Addl Field 1	Additional field for future use	O	Character	100
6	Addl Field 2	Additional field for future use	O	Character	100
7	Addl Field 3	Additional field for future use	O	Character	100
8	Addl Field 4	Additional field for future use	O	Character	100
9	Addl Field 5	Additional field for future use	O	Character	100
10	Addl Field 6	Additional field for future use	O	Character	100
11	Addl Field 7	Additional field for future use	O	Character	100
12	Addl Field 8	Additional field for future use	O	Character	100

2.4 Refund Transactions

Transaction Refund API facilitates the Merchant to process refund of transaction amount to the customer through online API. Refund amount can be either Partial or Full.

Refund of the transaction amount is possible **ONLY** for a settled transaction.

Request Parameters:

Sr. No.	Fields	Description	Mandatory(M) / Optional(O)	Data Type	Max Length
1	MID	Merchant ID	M	Character	15
2	Order ID	Unique Order Reference sent by Merchant	M	Character	30
3	PG Transaction Reference No	PG Transaction Reference No	M	Number	16
4	Transaction Amount	Amount to be refunded (**Paisa** format)	M	Number	15
5	Addl Field 1	Additional field for future use	O	Character	100
6	Addl Field 2	Additional field for future use	O	Character	100
7	Addl Field 3	Additional field for future use	O	Character	100
8	Addl Field 4	Additional field for future use	O	Character	100
9	Addl Field 5	Additional field for future use	O	Character	100
10	Addl Field 6	Additional field for future use	O	Character	100
11	Addl Field 7	Additional field for future use	O	Character	100
12	Addl Field 8	Additional field for future use	O	Character	100

Sample code to call Transaction Refund Request:

```
//Step 1: include the following namespace
<?php
include 'AWLMEAPI.php';

$obj = new AWLMEAPI();
$reqMsgDTO = new ReqMsgDTO();

$ reqMsgDTO ->setOrderId ( "1000001");
$reqMsgDTO->setMid( "AWL000000000001");
$reqMsgDTO->setRefundAmt(1000);  //Paisa Format
$reqMsgDTO->setPgMeTrnRefNo("21234567");
$reqMsgDTO->setEnckey("4f5390bey3ef1ee3d4a7e77fd42238cb");


//Step 3: Construct the request DTO with respective Parameter
$resMsgDTO = $obj->refundTransaction($reqMsgDTO);


//Step 4: Retrieve Status:
if ($resMsgDTO->getStatusCode()== "S")
        //Success
else
        //Failed
?>
```

Response Parameters:

Sr. No.	Fields	Description	Mandatory(M) / Optional(O)	Data Type	Max Length
1	Order ID	Unique Order Reference sent by Merchant	M	Character	30
2	PG Transaction Reference No.	PG Transaction Reference No.	M	Number	16
3	Refund Amount	Refund transaction amount(**Paisa** format)	M	Number	15
4	Status Code	Refund Request status(**S**-Success/**F**-Fail)	M	Character	1
5	Status Description	Refund Request status description	M	Character	100
6	Addl Field 1	Additional field for future use	O	Character	100

7	Addl Field 2	Additional field for future use	O	Character	100
8	Addl Field 3	Additional field for future use	O	Character	100
9	Addl Field 4	Additional field for future use	O	Character	100
10	Addl Field 5	Additional field for future use	O	Character	100
11	Addl Field 6	Additional field for future use	O	Character	100
12	Addl Field 7	Additional field for future use	O	Character	100
13	Addl Field 8	Additional field for future use	O	Character	100

3 API URL Configuration

Section 3.1:

The URL's that needs to be called for each of the API call is defined/configured in a properties file named **"ClientAPI.ini"**, which is available in the Kit. Below are the URL's that needs to be configured for each of the API calls:

Test Environment		
API	**Property**	**URL**
Transaction Status API	GET_TRANS_STATUS	https://cgt.in.worldline.com/ipg/getTransactionStatus
Cancel Transaction API	CANCEL_TRANS_API	https://cgt.in.worldline.com/ipg/doCancelRequest
Refund Transaction API	REFUND_TRANS_API	https://cgt.in.worldline.com/ipg/doRefundRequest

Production Environment		
API	**Property**	**URL**
Transaction Status API	GET_TRANS_STATUS	https://ipg.in.worldline.com/getTransactionStatus
Cancel Transaction API	CANCEL_TRANS_API	https://ipg.in.worldline.com/doCancelRequest
Refund Transaction API	REFUND_TRANS_API	https://ipg.in.worldline.com/doRefundRequest

Section 3.2:

Below is the URL that needs to be called when customer is redirected to WL PG Pay Page. This needs to be defined in the action property while doing the form submission.

Test Environment	
Form Action URL	https://cgt.in.worldline.com/ipg/doMEPayRequestCard

Production Environment	
Form Action URL	https://ipg.in.worldline.com/doMEPayRequestCard

4 Certificate

To ensure proper WL PG connectivity via PHP, required WL PG certificate needs to be included in the KIT during Integration.

Certificates for Test & Production environment are currently available in the "**Certificates**" folder. During Integration, based on the environment, it needs to be included in the Kit.

Certificate Name is as below:

- EntrustRootCertificationAuthority-G2.crt