ASTROPAY CARD

ASTROPAYCARD™

Version 2.3.1 English (technical)

CONTENTS TABLE

CONTENTS TABLE

1 Introduction

AstroPay Card is a virtual pre-paid card, accepted at hundreds of online stores integrated with AstroPay. Simply register for free, choose the value of the card you would like to purchase and pay using the most popular payments methods in your country. It's as easy as that!

AstroPay Card can be purchased in the 10 countries where AstroPay has operations. It can be paid through cash in thousands of payment stores, in can be paid through Boleto in Brazil, through Paysafecard in Argentina, bank transfer in the entire region, debit card and many more.

It is also instant, no need to wait. Simply purchase your card and you will immediately receive the card information on your email ready to be used in the site of your preference.

1.1 AboutAstroPay

AstroPay is the fastest growing payment solution in Latin America. It provides guaranteed payment solutions to consumers, avoiding issues like payment rejections, local taxes, fraud and credit approval. AstroPay also allows consumers to purchase online at international websites, using the most effective local payment methods in local currency.

AstroPay, a UK company, covers all the major markets in the Latin American region, providing payment solutions in markets where most popular solutions are not available, or not functioning properly. We pride ourselves on our ease of use, simple integration and operation for our merchants; easy registration and usability for our customers.

AstroPay is already the payment method of choice for millions of Latin American consumers.

1.2 Benefits of use AstroPay

1.2.1For the Merchants

Unless merchants accept local payment methods, the approval and conversion rates in Latin America will be minimal. AstroPay helps you raise merchants income rates with a first-class payment solution. AstroPay already processes large incomes for internet companies and can even be a merchant of record if required.

1.2.2For the Customers

Customers will want to find an easy, fast and secure payment solution in order to complete a deposit or payment in local currency avoiding the fact of using bank accounts revealing their personal and financial information to the sites.

1.2.3Features

1.2.3.1 Reliable

It's easy and free to register. No hidden fees. Customers feel at home by seamlessly paying online, at the bank or at payments stores through a secure, easy-to-understand platform.

Already used by the likes of Pokerstars& Expedia, AstroPay is a well-established company that processes millions of dollars in transactions, each month.

1.2.3.2 Instant

Users need fast payments processing to be quick and painless. AstroPay's instant confirmation is perfect for digital services and products which customers want to use right after paying.

1.2.3.3 Compliant

AstroPay is integrated with the most popular payment methods in every country they run operations at. Customers can use their preferred payment methods (Boleto, Online Bank Payments, Debit Cards, Cash payments) and pay in their local currency. Besides they are fully compliant with each country banking regulations.

1.2.3.4 Global

Take advantage of other markets without needing to expand operations. AstroPay runs operations in the entire LATAM region with presence in the following countries: Argentina, Brazil, Mexico, Chile, Colombia, Peru, Bolivia, Uruguay, Paraguay, Costa Rica, and Venezuela. It also offers customer support in English, Spanish and Portuguese.

1.2.3.5 Safe & Secure

AstroPay's solution is compliant with anti-money laundering laws and offers both customers and merchants complete security in their transactions.

Customers do not have to expose their financial and personal information. Payments are 100% guaranteed. No chargeback, no hassle. Since it's a prepaid payments solution there is no risk on chargeback for merchants neither overspendings for customers.

1.3 How it works

2 Integration

2.1 Integration Kit Contents

- Integration manual
- SDK
  - AstroPayCard.class.php (AstroPay Card class)
  - sample.php
- Errors table document

2.2 ¿How I should integrate AstroPay?

For correct integration of AstroPay Card, we recommend follow orderly the following steps:

1. Configure your credentials (provided by AstroPay Support team) within the AstroPayCard.class.php class provided in this Integration KIT.
2. Execute the sample in your server, for better learning we recommend that execute all methods provided by the AstroPayCard class. (See section 2.1 for more info)
3. Study all functions in the section 3 “API structure”.
4. Implement the solution in your software using the knowledge acquired in previous tests using the Sandbox system.
5. Verify the integration.
6. Once you verified your integration, contact with techsupport@astropaycard.com to homologate your integration and put the system in production
7. Enjoy AstroPay service!

2.3 General considerations

- Communication between the merchant and ASTROPAY be made by POST.
- The protocol used is purely HTTPS, without using SOAP.
- The merchant’scredentials also need to be sent by POST without any special attributes, for example, SOAP headers.

2.4 Sample code

In the sample code you can try how the AstroPay API works. It’s much recommended to try first with these samples.

You can found it in **SDK/sample.php** (the SDK provided by AstroPay).

2.5 String control MD5 hash for return post

Please note: the periods / full-stops in the following codes mean concatenated.

Consider the following variables:

- `$x_login` – Merchant login id credential. Provided by AstroPay.
- $transaction_id – Transaction ID provided by AstroPay.
- $amount – The amount of the transaction

The control string is calculated:

```
$control = md5($x_login . $transaction_id . $amount);
```

3 AstroPay servers

3.1 Production

3.1.1 Server information

URL: https://api.astropaycard.com/verif/validator
IP: 54.229.22.192

3.2 Sandbox

The sandbox system provides a testing system that allow integrate AstroPay Card service using fictitious money.

We recommend to do the integration using the sandbox first, and then change the AstroPayCard class settings to use the production system.

3.2.1 Server information

URL: https://sandbox-api.astropaycard.com/verif/validator
IP: 54.229.12.81

3.2.2 Test data

3.2.2.1 Cards

$25 AstroPay Card
Number:1175000000000025
Security code (CVV): 1234
Expiration date: 05/2015
Value: $25

$100AstroPay Card
Number:1175000000000100
Security code (CVV): 1234
Expiration date: 05/2015
Value: $100

$500AstroPay Card
Number:1175000000000500
Security code (CVV): 1234
Expiration date: 05/2015
Value: $500

InvalidAstroPay Card
Number:1175000000000025
Security code (CVV): 1254
Expiration date: 05/2015
Value: $25

4 API structure

This is a description of the AstroPayCard API structure. Please read it, and refer it when you have doubts in the integration.

4.1 /verif/validator.php

4.1.1Whatisit?

Basically, the /verif/validator is the WebService that recieve and process the transaction requests.

You're going to use this WebService to:

- Authorize transactions (x_type=AUTH_ONLY)
- Capture transactions (x_type=CAPTURE_ONLY)
- Authorize and capture transactions (x_type=AUTH_CAPTURE)
- Void transactions (x_type=VOID)
- Refund transactions (x_type=REFUND)

*Note: The **x_type** is a parameter that you will use in API calls.*

> **Important:**
> The transactions sent with the same amount, and card information, will taken as duplicated with a time window of 120 seconds.
> So, we **don't recommend** doing **batch transactions**.

4.1.2Call

URL: **https://api.astropaycard.com/verif/validator**

4.1.2.1 Parameterstable

Field	Description	Type	Mandatory	Example	Default value
x_login	Merchant login id credential. Provided by AstroPay	String(15)	Yes	fnWf39Sa1Q	-

x_tran_key	Merchant trans key credential. Provided by AstroPay	String(15)	Yes	Hiw35VsFrw	-
x_version	API version.	Decimal(4,2)	Yes	2.0	-
x_type	The transaction type.	string. Options: (“AUTH_ONLY”, “CAPTURE_ONLY”, “AUTH_CAPTURE”, “CREDIT”, “REFUND”, “VOID”)	Yes	AUTH_CAPTURE	-
x_test_request	Selects if the transaction is a test	Char(1)	Yes	Y	-
x_card_num	AstroPay Card number	string(25)	Yes	1145426000818586	-
x_card_code	AstroPayc Card security code (CVV)	integer	Yes	1234	-
x_exp_date	Expiration date of AstroPay Card	string(7)	Yes	02/2013	-
x_amount	Transaction amount	decimal(10,2)	Yes	25.69	-
x_unique_id	Unique, anonymized identifier of users in themerchant system	string(50)	Yes	abs1522566	-
x_invoice_num	Unique identifier of Merchant transaction	string(50)	Yes	dsa3jg4638n	-

x_duplicate_window	The time window after a transaction is taken as duplicated	integer	No	60	120
x_delim_char	Response delimiting char between fields when the x_response_format is “string”	Char(1)	No	@	|
x_response_format	The response format. Options: “string”, “json”, “xml”	String	No	json	string
x_delim_data	The request to receive a delimited transaction response	boolean	No	true	True
x_method	Type of pay method	string. Options: “CC”	No	CC	CC
x_trans_id	(**For VOID, CREDIT and REFUND only**) TransactionID param returned by AstroPay on AUTH_CAPTURE or CAPTURE_ONLY previous tranasction	string(50)	Yes (Only for VOID, CREDIT, REFUND)	123456	-
x_auth_code	(**For CAPTURE_ONLY only**)	string(20)	Yes (Only for CAPTURE_ONLY)	as3gfd42d	-

	TransactionID param sent by AstroPay in previous AUTH_ONLY call

4.1.2.2 Additional parameters

You can send us more information for reference proporses:

Transaction
x_description (product description)

About your customer
x_cust_id
x_first_name
x_last_name
x_company
x_address
x_city
x_state
x_zip
x_country
x_phone
x_fax
x_email

Shipping information
x_ship_to_first_name
x_ship_to_last_name
x_ship_to_company
x_ship_to_address
x_ship_to_city
x_ship_to_state
x_ship_to_zip
x_ship_to_country

Tax information
x_tax
x_duty
x_freight
x_tax_exempt
Order number
x_po_num

*Note: You can add any more information with other parameter but it **NOT** be begin with "x_".*

4.1.2.3 Example

URL: https://api.astropaycard.com/verif/validator
Params:
x_login=abs35nnl4&x_tran_key=bfkd4jBe2&x_version=2.0&x_type=AUTH_CAPTURE&x_test_request=N&x_card_num=1145426000818586&x_card_code=1234&x_exp_date=05/2013&x_amount=12.03&x_unique_id=user9852&x_invoice_num=trans-123456

4.1.3Response

The response can get the format that you want between "json", "string" and "xml". Is recommended use "json" or "xml" for better data integrity.

Field	Description	Type	Example
response_code	Transaction response code 1-Approved 2-Declined 3-Error	integer	1
response_subcode	Subcode response (see response codes table)	integer	1
response_reason_code	(See response codestable)	integer	1
response_reason_text	Description of the transaction result	string(200)	"Transaction OK!"
approval_code	Authrizationcode	string(50)	"abc123"
AVS	AVS of thetransaction	char(1)	"P"
TransactionID	Transaction ID provided by AstroPay	integer	123456
r_unique_id	Astropay user who's using thecard (Numeric) (Only for version 2.0 or greater	integer	321654
x_invoice_num	Invoice number provided by merchant	string(50)	invoice-12345
x_description	Description of the transaction provided	string(200)	Transactiondescriptionblabla..

	by the merchant		
x_amount	The amount of the transaction. ***Important:*** *For safety reasons, it's highly recommended compare the amount sent with the response amount*	decimal(10,2)	20.6
x_method			
x_type	Transaction type (AUTH_CAPTURE, etc.)	string. Options: ("AUTH_ONLY", "CAPTURE_ONLY", "AUTH_CAPTURE", "CREDIT", "REFUND", "VOID")	AUTH_CAPTURE
x_cust_id	(For merchant proporse)	string	abcabc...
x_first_name	(For merchant proporse)	string	abcabc...
x_last_name	(Formerchantproporse)	string	abcabc...
x_company	(Formerchantproporse)	string	abcabc...
x_address	(Formerchantproporse)	string	abcabc...
x_city	(Formerchantproporse)	string	abcabc...
x_state	(Formerchantproporse)	string	abcabc...
x_zip	(Formerchantproporse)	string	abcabc...
x_country	(Formerchantproporse)	string	abcabc...
x_phone	(Formerchantproporse)	string	abcabc...
x_fax	(Formerchantproporse)	string	abcabc...
x_email	(Formerchantproporse)	string	abcabc...

x_ship_to_first_name	(Formerchantpropose)	string	abcabc…
x_ship_to_last_name	(Formerchantpropose)	string	abcabc…
x_ship_to_company	(Formerchantpropose)	string	abcabc…
x_ship_to_address	(Formerchantpropose)	string	abcabc…
x_ship_to_city	(Formerchantpropose)	string	abcabc…
x_ship_to_state	(Formerchantpropose)	string	abcabc…
x_ship_to_zip	(Formerchantpropose)	string	abcabc…
x_ship_to_country	(Formerchantpropose)	string	abcabc…
x_tax	(Formerchantpropose)	string	abcabc…
x_duty	(Formerchantpropose)	string	abcabc…
x_freight	(Formerchantpropose)	string	abcabc…
x_tax_exempt	(Formerchantpropose)	string	abcabc…
x_po_num	(Formerchantpropose)	string	abcabc…
x_test_request	Indicateswhether the transaction is a test or not	string Options: (“Y”, “N”)	Y
md5_hash	Generated MD5 hash value that may be used to authenticate thetransaction response (see 2.2 String control MD5 hash for return post)	string	dsgtdfy515arwgfd
cc_response	The card code verification (CCV) response code M = Match N = No Match P = Not Processed S = Should have	string. Options: (“M”, “N”, “P”, ”S”, ”U”)	M

	been present U = Issuer unable to process request
reserved40-66	All other information that the merchant sends

4.2 /verif/transtatus

4.2.1What is it?

This function will provide information about transactions in any moment.

You may use this function to consult a single transaction status in a particular case.

4.2.2Call

URL: **https://api.astropaycard.com/verif/transtatus**

4.2.2.1 Parameters table

Field	Description	Type	Mandatory	Example	Default value
x_login	Merchant login id credential. Provided by AstroPay	String(15)	Yes	fnWf39Sa1Q	-
x_tran_key	Merchant trans key credential. Provided by AstroPay	String(15)	Yes	Hiw35VsFrw	-
x_invoice_num	Unique identifier of Merchant transaction, (Inserted before validator.php	string(50)	Yes	dsa3jg4638n	-

	call)				
x_test_request	Selects if the transaction was a test	Char(1)	Yes	Y	-
x_type	0: for simple information 1: for detailed information (see response section)	integer	Yes	AUTH_CAPTURE	-
x_delim_char	Response delimiting char between fields when the x_response_format is “string”	Char(1)	No	@	|

4.2.2.2 Example

*Note:*This example is an extension of the first example viewed in section 3.1.2.3 (validator example)

URL: https://api.astropaycard.com/verif/transtatus
Params:
x_login=abs35nnl4&x_tran_key=bfkd4jBe2&x_type=1&x_test_request=N&x_invoice_num=trans-123456

4.2.3Response

The response can get the format that you want between “json”, “string” and “xml”. We recommend use “json” or “xml” for best data integrity.

Field	Description	Type	Example
code	Transaction response code 1-Approved 2-Declined 3-Error	integer	1
subcode	Subcode response	integer	1

	(see response codes table)		
reason_code	(See response codestable)	integer	1
reason_text	Description of the transaction result	string(200)	“Transaction OK!”
txn_date	Date of the transaction	string(20)	“2013-03-01 21:26:31”
txn_type			
x_amount	Transaction amount	decimal(10,2)	“20.36”
TransactionID	AstroPay transaction unique ID	integer	523456
x_invoice_num	Merchant transaction unique ID	integer	trans-123456
x_description	Transaction description provided by Merchant	string(255)	“IT magazine Subscription”
r_result			
r_authorization_code	Authrizationcode	string(50)	“abc123”
process_status			
md5hash	Response MD5 hash. May be used to authenticate the transaction response	string	dsgtdfy515arwgfd

5 AstroPay Card integration class

5.1 What is it?

This class is an abstraction of the AstroPay Card API. It will help you with the integration and provides an homogeneous interface to ensure an correct integration.

The class can be seen and studied here.

> **Important:**
> The configuration and method parameters are the same (name, type and function) that described in sections Parameterstable and Parameters table.

5.2 Configuration

You have to change the class configuration to assign your credentials and personalize some aspects.

To modify them, you must change the class attributes on the beginning of the class.

5.2.1 Credentials

Atributes**x_login** and **x_trans_key** are the credentials to configure. Please contact with AstroPay (techsupport@astropaycard.com) and we provide them.

5.2.2 General settings

In the general settings you can change some aspects of the API behavior. Please refer sections 3.1.2.1 and 3.2.2.1 to see the full descriptions.

5.2.3 Sandbox mode

In the integration stage, you should use the sandbox server, that is the same system as production, but you don't use real money.

To set the sandbox mode on the class, you should set the $sandbox attribute to true.

Code (AstroPayCard.class.php)

```
//Sandbox (TODO: Change to false in production)
private $sandbox = true;
```

5.3 Methods

AstroPayCardclass provide the following methods:

- auth_transaction (equivalent AUTH_ONLY transaction type)
- capture_transaction (equivalent CAPTURE_ONLY transaction type)
- auth_capture_transaction (equivalent AUTH_CAPTURE transaction type)
- check_transaction_status (equivalent to /verif/transtatus call)
- refund_transaction (equivalent REFUND transaction type)
- void_transaction (equivalent VOID transaction type)

5.3.1 auth_transaction and capture_transaction

This method will make an authorization, returning whether the user can pay the transaction amount or not and reserve the amount.

5.3.1.1 Params

- x_card_num: The AstroPay Card number.
- x_card_code: The AstroPay Card CVV.
- x_exp_date: The AstroPay Card expiration date.
- x_amount: The amount of the transaction
- x_unique_id: The user id of the merchant
- x_invoice_num: The invoice number of the merchant. This ID will be generated by the Merchant.

5.3.1.2 Usage example

Usage example:

```
$astropay = new AstroPayCard();
//NOTE: Use json_decode only if the x_response_formatconfigparam is set in “json”

//**** Auth first, and capture after ****//
//Authorize the transaction
$response = json_decode($astropay->auth_transaction($x_card_num, $x_card_code,
$x_exp_date, $x_amount, $x_unique_id, $x_invoice_num));
$approval_code = $response->approval_code;

//Capture transaction using the approval_code returned by authorization
$final_response = json_decode($astropay->capture_transaction($approval_code,
$x_card_num, $x_card_code, $x_exp_date, $x_amount, $x_unique_id,
$x_invoice_num));
```

5.3.1.3 Return

The return will be a string of the parameters shown in section 3.1.3 with the x_response_format (string, json or xml) sent. By default the response format will be a pipe separated string.

5.3.2auth_capture_transaction

This method will make an authorization and capture in the same call.

5.3.2.1 Params

- x_card_num: The AstroPay Card number.
- x_card_code: The AstroPay Card CVV.
- x_exp_date: The AstroPay Card expiration date.
- x_amount: The amount of the transaction
- x_unique_id: The user id of the merchant
- x_invoice_num: The invoice number of the merchant. This ID will be generated by the Merchant.

5.3.2.2 Usage example

Usage example:

```
$astropay = new AstroPayCard();

//**** Auth capture ****//
$response = $astropay->auth_capture_transaction($x_card_num, $x_card_code,
$x_exp_date, $x_amount, $x_unique_id, $x_invoice_num);
```

5.3.2.3 Return

The return will be a string of the parameters shown in section 3.1.3 with the x_response_format (string, json or xml) sent. By default the response format will be a pipe separated string.

5.3.3auth_capture_transaction

This method willmake an authorization and capture in the same call.

5.3.3.1 Params

- x_card_num: The AstroPay Card number.
- x_card_code: The AstroPay Card CVV.
- x_exp_date: The AstroPay Card expiration date.

- x_amount: The amount of the transaction
- x_unique_id: The user id of the merchant
- x_invoice_num: The invoice number of the merchant. This ID will be generated by the Merchant.

5.3.3.2 Usage example

Usage example:

```
$astropay = new AstroPayCard();

//**** Auth capture ****//
$response = $astropay->auth_capture_transaction($x_card_num, $x_card_code,
$x_exp_date, $x_amount, $x_unique_id, $x_invoice_num);
```

5.3.3.3 Return

The return will be a string of the parameters shown in section 3.1.3 with the x_response_format (string, json or xml) sent. By default the response format will be a pipe separated string.

5.3.4check_transaction_status

This method will return the transaction status of the x_invoice number provided.

You may use them to check a transaction status in any moment.

5.3.4.1 Params

- x_invoice_num: The invoice number of the merchant. This ID should be the merchant id used in the original transaction to check.

5.3.4.2 Usage example

Usage example:

```
$astropay = new AstroPayCard();

//**** Check transaction status ****//
$response = $astropay->check_transaction_status($x_invoice_num);
```

5.3.4.3 Return

The return will be a string of the parameters shown in section 3.2.3 with the x_response_format (string, json or xml) sent. By default the response format will be a pipe separated string.

5.3.5refund_transaction

This method will refund a transaction.

You may use them to refund a settled transaction referenced by the TransactionIDparam sent by AstroPay in the original transaction.

5.3.5.1 Params

- transaction_id: The TransactionIDparam returned by AstroPay in the original transaction.
- x_card_num: The AstroPay Card number.
- x_card_code: The AstroPay Card CVV.
- x_exp_date: The AstroPay Card expiration date.
- x_amount: The amount of the transaction

5.3.5.2 Usage example

Usage example:

```
$astropay = new AstroPayCard();

//**** Refund ****//
$response = $astropay->refund_transaction($transaction_id, $x_card_num,
$x_card_code, $x_exp_date, $x_amount);
```

5.3.5.3 Return

The return will be a string of the parameters shown in section 3.1.3 with the x_response_format (string, json or xml) sent. By default the response format will be a pipe separated string.

5.3.6void_transaction

This method will void a transaction.

You may use them to void an unsettled transaction referenced by the TransactionIDparam sent by AstroPay in the original transaction.

5.3.6.1 Params

- transaction_id: The TransactionIDparam returned by AstroPay in the original transaction.
- x_card_num: The AstroPay Card number.
- x_card_code: The AstroPay Card CVV.
- x_exp_date: The AstroPay Card expiration date.
- x_amount: The amount of the transaction

5.3.6.2 Usage example

Usage example:

```
$astropay = new AstroPayCard();

//**** Void ****//
$response = $astropay->void_transaction($transaction_id, $x_card_num,
$x_card_code, $x_exp_date, $x_amount);
```

5.3.6.3 Return

The return will be a string of the parameters shown in section 3.1.3 with the x_response_format (string, json or xml) sent. By default the response format will be a pipe separated string.

6 Security details

- The messages will require a username and a password
- An HMAC-SHA-256 (RFC 2104) code will be used to verify the integrity of the information received from the merchant.
- All communications between the merchant and ASTROPAY will be made exclusively through registered IPs on both sides.