
ATLPay API Documentation Documentation

Release 1.0.0

FINPAYTECH GLOBAL SOLUTIONS

Apr 08, 2020

Table of Contents:

1	Errors	3
1.1	Error Codes	3
1.2	Error Attributes	4
2	Tokens	5
2.1	The token object	6
2.2	Create a card token	6
2.3	Retrieve a token	7
3	Charges	9
3.1	The charge object	9
3.2	Create a charge	10
3.3	Retrieve a charge	11
3.4	Capture a charge	12
4	Refunds	15
4.1	The refund object	15
4.2	Create a refund	15
4.3	Retrieve a refund	16
5	Introduction	17
6	Indices and tables	19

Authenticate your account by including your secret key in API requests. You can manage your API keys in the Dashboard. Your API keys carry many privileges, so be sure to keep them secure! Do not share your secret API keys in publicly accessible areas such as GitHub, client-side code, and so forth.

To use your API key, you need only call `\ATLPay\ATLPay::setSecretKey()` with your key. The PHP library will then automatically send this key in each request.

cURL Example

```
curl -X POST \
  https://api.atlpay.com/v2/tokens \
  -H 'X-API-Key: YOUR_API_KEY_HERE'
```

All API requests must be made over HTTPS. Calls made over plain HTTP will fail. API requests without authentication will also fail.

ATLPay uses conventional HTTP response codes to indicate the success or failure of an API request. In general: Codes in the 2xx range indicate success. Codes in the 4xx range indicate an error that failed given the information provided (e.g., a required parameter was omitted, a charge failed, etc.). Codes in the 5xx range indicate an error with ATLPay's servers (these are rare).

Some 4xx errors that could be handled programmatically (e.g., a card is declined) include an error code that briefly explains the error reported.

1.1 Error Codes

HTTP Code	Description
200 - OK	Everything worked as expected.
400 - Bad Request	The request was unacceptable, often due to missing a required parameter.
401 - Unauthorized	No valid API key provided.
402 - Request Failed	The parameters were valid but the request failed.
403 - Forbidden	You cannot access this resource.
404 - Not Found	The requested resource doesn't exist.
500, 502, 503, 504 - Server Errors	Something went wrong on ATLPay's end. (These are rare.)

1.2 Error Attributes

Attribute	Description
<code>type string</code>	The type of error returned
<code>message string</code>	A human-readable message providing more details about the error.
<code>error.code string</code>	For some errors that could be handled programmatically, a short string indicating the error code reported.
<code>error.message string</code>	A human-readable message providing more details about the error.
<code>error.parameter string</code>	if the error is parameter-specific, the parameter related to the error. For example, you can use this to display a message near the correct form field.

Tokenization is the process ATLPay uses to collect sensitive card or bank account details, or personally identifiable information (PII), directly from your customers in a secure manner. A token representing this information is returned to your server to use. You should use [ATLPay.js](#) or our mobile libraries to perform this process, client-side. This ensures that no sensitive card data touches your server, and allows your integration to operate in a PCI-compliant way.

If you cannot use client-side tokenization, you can also create tokens using the API with your secret API key. Keep in mind that if your integration uses this method, you are responsible for any PCI compliance that may be required, and you must keep your secret API key safe. Unlike with client-side tokenization, your customer's information is not sent directly to ATLPay, so we cannot determine how it is handled or stored.

Tokens cannot be stored or used more than once.

API Endpoint : `https://api.atlpay.com/v2/tokens`

2.1 The token object

Attribute	Description
type	type of the object, in this case it will return <code>token</code>
id	Unique identifier for the object.
card.fundingType	Funding type of card <code>CREDIT</code> or <code>DEBIT</code>
card.country	ISO 2 country code of the card eg. <code>GB</code> , <code>FR</code> etc.
card.last4Digits	Last 4 digits of the card
card.type	Type of the card eg. <code>VISA_DEBIT</code> OR <code>VISA_CREDIT</code>
card.brand	Brand of the card eg. <code>Visa</code> , <code>MasterCard</code> etc.
card.bank	Issuer Bank of the Card
card.name	Cardholder name
card.authorization	If additional authorization is required or not. Please refer to 3-D Security for more details.
card.addressLine1	Billing address line 1
card.addressLine2	Billing address line 2
card.addressCity	Billing city
card.addressState	Billing state/region/county
card.addressZip	Billing zipcode or postal code
card.addressCountry	ISO 2 country code of billing country
created	Time at which the object was created. Measured in seconds since the Unix epoch.
mode	<code>live</code> if the object exists in live mode or the value <code>test</code> if the object exists in test mode.

2.2 Create a card token

Creates a single-use token that represents a credit card's details. This token can be used in place of a credit card dictionary with any API method. These tokens can be used only once.

In most cases, you should create tokens client-side using `ATLPay.js` or our mobile libraries, instead of using the API.

Attribute	Mandatory	Description
card[number]	Yes	The card number, as a string without any separators.
card[cvc]	Yes	Card security code.
card[exp_month]	Yes	Two-digit number representing the card's expiration month.
card[exp_year]	Yes	Four-digit number representing the card's expiration year.
card[name]	Yes	Cardholder's full name.
address[address_line1]	No	Billing address line 1
address[address_line2]	No	Billing address line 2
address[city]	No	Billing city
address[state]	No	Billing state/region/county
address[zipcode]	No	Billing zipcode or postal code
address[country]	No	ISO 2 country code of billing country
shopper[ip]	Yes	IP Address of customer
shopper[session_id]	Yes	Session ID of customer online session
shopper[email]	Yes	Email address of the customer

Returns

Returns the created card token if successful. Otherwise, this call returns an error.

Example request

```
curl -X POST \
  https://api.atlpay.com/v2/tokens \
  -H 'X-API-Key: YOUR_API_KEY' \
  -F 'card[number]=5555555555554444' \
  -F 'card[cvc]=938' \
  -F 'card[exp_month]=08' \
  -F 'card[exp_year]=2020' \
  -F 'card[name]=USER NAME' \
  -F 'shopper[ip]=203.163.245.135' \
  -F 'shopper[session_id]=sess_12312asdhasd7' \
  -F 'shopper[email]=user@example.com'
```

Example response

```
{
  "type": "token",
  "id": "776e05d1-cda5-4f78-8d36-761a23b8a30a",
  "card": {
    "fundingType": "DEBIT",
    "country": "GB",
    "last4Digits": "4444",
    "type": "VISA_DEBIT",
    "brand": "VISA",
    "bank": "BANK OF IRELAND (UK) PLC",
    "name": "DEMO USER",
    "authorization": "REQUIRED",
    "addressLine1": null,
    "addressLine2": null,
    "addressCity": null,
    "addressState": null,
    "addressZip": null,
    "addressCountry": null
  },
  "created": 1530260981,
  "mode": "live"
}
```

2.3 Retrieve a token

Returns a token if a valid ID was provided. Returns an error otherwise.

Example request

```
curl -X GET \
  https://api.atlpay.com/v2/tokens/776e05d1-cda5-4f78-8d36-761a23b8a30a \
  -H 'X-API-Key: YOUR_API_KEY'
```

Example response

```
{
  "type": "token",
  "id": "776e05d1-cda5-4f78-8d36-761a23b8a30a",
  "card": {
    "fundingType": "DEBIT",
    "country": "GB",
```

(continues on next page)

(continued from previous page)

```
    "last4Digits": "4444",
    "type": "VISA_DEBIT",
    "brand": "VISA",
    "bank": "BANK OF IRELAND (UK) PLC",
    "name": "DEMO USER",
    "authorization": "REQUIRED",
    "addressLine1": null,
    "addressLine2": null,
    "addressCity": null,
    "addressState": null,
    "addressZip": null,
    "addressCountry": null
  },
  "created": 1530260981,
  "mode": "live"
}
```

Charges

To charge a credit or a debit card, you create a `Charge` object. You can retrieve and refund individual charges. Charges are identified by a unique, random ID.

API Endpoint : <https://api.atlpay.com/v2/charges>

3.1 The charge object

Attribute	Description
<code>type</code>	type of the object, in this case it will return <code>charge</code>
<code>id</code>	Unique identifier for the object.
<code>currency</code>	Three-letter ISO currency code, in lowercase. Must be a supported currency.
<code>amount</code>	A positive integer in the smallest currency unit (e.g., 100 to charge £1.00) representing how much to charge.
<code>fees</code>	The fee (if any) for the charge.
<code>description</code>	An arbitrary string attached to the object. Often useful for displaying to users.
<code>txn_reference</code>	ID of the invoice this charge is for if one exists.
<code>threeSecure.redirect</code>	The URL provided to you to redirect a customer to as part of a redirect authentication flow.
<code>threeSecure.returnUrl</code>	The URL you provide to redirect the customer to after they authenticated their payment.
<code>threeSecure.status</code>	The status of the redirect, either <code>PENDING</code> (ready to be used by your customer to authenticate the transaction), <code>CHARGEABLE</code> (successful authentication, cannot be reused) or <code>NOT_AVAILABLE</code> (redirect should not be used) or <code>FAILED</code> (failed authentication, cannot be reused).
<code>status</code>	Status of the charge
<code>token</code>	The token object used for charge
<code>created</code>	Time at which the object was created. Measured in seconds since the Unix epoch.
<code>mode</code>	<code>live</code> if the object exists in live mode or the value <code>test</code> if the object exists in test mode.

3.2 Create a charge

To charge a credit card or other payment source, you create a Charge object. If your API key is in test mode, the supplied payment source (e.g., card) won't actually be charged, although everything else will occur as if in live mode. (ATLPay assumes that the charge would have completed successfully).

Attribute	Mandatory	Description
token	Yes	A card token to be charged, like the ones returned by ATLPay.js .
amount	Yes	A positive integer in the smallest currency unit (e.g., 100 to charge £1.00) representing how much to charge.
currency	Yes	3-letter ISO code for currency.
description	Yes	An arbitrary string which you can attach to a Charge object. It is displayed when in the web interface alongside the charge.
txn_reference	No	ID of the invoice this charge is for if one exists.
return_url	YES	The URL you provide to redirect the customer to after they authenticated their payment.

Returns

Returns a Charge object if the charge succeeded. Returns an error if something goes wrong. A common source of error is an invalid or expired card, or a valid card with insufficient available balance.

Example request

```
curl -X POST \
  https://api.atlpay.com/v2/charges \
  -H 'X-API-Key: YOUR_API_KEY' \
  -F token=776e05d1-cda5-4f78-8d36-761a23b8a30a \
  -F amount=500 \
  -F currency=GBP \
  -F 'description=Order Desc' \
  -F 'txn_reference=Your Order Number' \
  -F return_url=https://www.your-3d-return-url.com/
```

Example response

```
{
  "type": "charge",
  "id": "C2018062972567",
  "currency": "GBP",
  "amount": 500,
  "fees": null,
  "description": "Order Desc",
  "txn_reference": "Your Order Number",
  "threeSecure": {
    "redirect": "https://payments.atlpay.com/3d-secure/C2018062972567?
↪paRes=MjU3NDQyM2YtNmFmMy00OTEzLWI1YTUtZmZjY2EzODg4ZGYx",
    "returnUrl": "https://www.your-3d-return-url.com/",
    "status": "PENDING"
  },
  "status": "PENDING",
  "token": {
    "type": "token",
    "id": "776e05d1-cda5-4f78-8d36-761a23b8a30a",
    "card": {
      "fundingType": "DEBIT",
```

(continues on next page)

(continued from previous page)

```

    "country": "GB",
    "last4Digits": "3005",
    "type": "VISA_DEBIT",
    "brand": "VISA",
    "bank": "BANK OF IRELAND (UK) PLC",
    "name": "Richard AMOAH",
    "authorization": "REQUIRED",
    "addressLine1": null,
    "addressLine2": null,
    "addressCity": null,
    "addressState": null,
    "addressZip": null,
    "addressCountry": null
  },
  "created": 1530260981
},
"created": 1530266033,
"mode": "live"
}

```

3.3 Retrieve a charge

Retrieves the details of a charge that has previously been created. Supply the unique charge ID that was returned from your previous request, and ATLPay will return the corresponding charge information. The same information is returned when creating or refunding the charge.

Example request

```

curl -X GET \
  https://api.atlpay.com/v2/charges/C2018062972567 \
  -H 'X-API-Key: YOUR_API_KEY'

```

Example response

```

{
  "type": "charge",
  "id": "C2018062972567",
  "currency": "GBP",
  "amount": 500,
  "fees": null,
  "description": "Order Desc",
  "txn_reference": "Your Order Number",
  "threeSecure": {
    "redirect": "https://payments.atlpay.com/3d-secure/C2018062972567?
    ↪paRes=MjU3NDQyM2YtNmFmMy00OTEzLWl1YTUtZmZjY2EzODg4ZGYx",
    "returnUrl": "https://www.your-3d-return-url.com/",
    "status": "PENDING"
  },
  "status": "PENDING",
  "token": {
    "type": "token",
    "id": "776e05d1-cda5-4f78-8d36-761a23b8a30a",
    "card": {
      "fundingType": "DEBIT",

```

(continues on next page)

(continued from previous page)

```

    "country": "GB",
    "last4Digits": "3005",
    "type": "VISA_DEBIT",
    "brand": "VISA",
    "bank": "BANK OF IRELAND (UK) PLC",
    "name": "Richard AMOAH",
    "authorization": "REQUIRED",
    "addressLine1": null,
    "addressLine2": null,
    "addressCity": null,
    "addressState": null,
    "addressZip": null,
    "addressCountry": null
  },
  "created": 1530260981
},
"created": 1530266033,
"mode": "live"
}

```

3.4 Capture a charge

Capture the payment of an existing, uncaptured, charge. Uncaptured payments expire exactly seven days after they are created. If they are not captured by that point in time, they will be marked as refunded and will no longer be capturable.

Returns

Returns the charge object. Capturing a charge will always succeed, unless the charge is already refunded, expired, captured in which case this method will return an error.

Example request

```

curl -X POST \
  https://api.atlpay.com/v2/charges/capture/C2018062972567 \
  -H 'X-API-Key: YOUR_API_KEY'

```

Example response

```

{
  "type": "charge",
  "id": "C2018062972567",
  "currency": "GBP",
  "amount": 500,
  "fees": 33,
  "description": "Order Desc",
  "txn_reference": "Your Order Number",
  "threeSecure": {
    "redirect": "https://payments.atlpay.com/3d-secure/C2018062972567?
↪paRes=MjU3NDQyM2YtNmFmMy00OTEzLWI1YTUtZmZjY2EzODg4ZGYx",
    "returnUrl": "https://www.your-3d-return-url.com/",
    "status": "CAPTURED"
  },
  "status": "CHARGE_SUCCESS",
  "token": {

```

(continues on next page)

(continued from previous page)

```
"type": "token",
"id": "776e05d1-cda5-4f78-8d36-761a23b8a30a",
"card": {
  "fundingType": "DEBIT",
  "country": "GB",
  "last4Digits": "3005",
  "type": "VISA_DEBIT",
  "brand": "VISA",
  "bank": "BANK OF IRELAND (UK) PLC",
  "name": "Richard AMOAH",
  "authorization": "REQUIRED",
  "addressLine1": null,
  "addressLine2": null,
  "addressCity": null,
  "addressState": null,
  "addressZip": null,
  "addressCountry": null
},
"created": 1530260981
},
"created": 1530269496,
"mode": "live"
}
```


To refund a payment, you create a `Refund` object. You can retrieve and process partial or full refund. Refunds are identified by a unique, random ID.

API Endpoint : `https://api.atlpay.com/v2/charges/refund/{CHARGE-ID-HERE}`

4.1 The refund object

Attribute	Description
<code>type</code>	type of the object, in this case it will return <code>refund</code>
<code>id</code>	Unique identifier for the object.
<code>currency</code>	Three-letter ISO currency code, in lowercase.
<code>amount</code>	A positive integer in the smallest currency unit (e.g., 100 to charge £1.00) representing refund amount.
<code>fee_refunded</code>	The fee refunded (if any)
<code>status</code>	Status of the refund
<code>created</code>	Time at which the object was created. Measured in seconds since the Unix epoch.
<code>mode</code>	<code>live</code> if the object exists in live mode or the value <code>test</code> if the object exists in test mode.

4.2 Create a refund

To refund a charge, you create a `Refund` object. If your API key is in test mode, then charge won't actually be refunded, although everything else will occur as if in live mode. (ATLPay assumes that the refund would have completed successfully).

Attribute	Mandatory	Description
<code>amount</code>	No	A positive integer in the smallest currency unit (e.g., 100 to charge £1.00) representing how much to refund. If not provided the full amount or remaining balance will be refunded

Returns

Returns a Charge object if the charge succeeded. Returns an error if something goes wrong. A common source of error is refunding an already refunded charge.

Example request

```
curl -X POST \  
  https://api.atlpay.com/v2/charges/refund/{CHARGE-ID-HERE} \  
  -H 'X-Api-Key: YOUR_API_KEY' \  
  -F amount=500 \  
  \
```

Example response

```
{  
  "type": "refund",  
  "id": "978dfd87-13b8-443a-892e-fca0380ef7d4",  
  "amount": 500,  
  "currency": "GBP",  
  "fee_refunded": 0,  
  "status": "REFUNDED",  
  "created": "2020-04-08T11:55:21",  
  "mode": "test"  
}
```

4.3 Retrieve a refund

Retrieves the details of a refund that has previously been created. Supply the unique charge ID that was returned from your previous request, and ATLPay will return the corresponding refund information. The same information is returned when creating a refund.

Example request

```
curl -X GET \  
  https://api.atlpay.com/v2/charges/refund/978dfd87-13b8-443a-892e-fca0380ef7d4 \  
  -H 'X-Api-Key: YOUR_API_KEY'
```

Example response

```
{  
  "type": "refund",  
  "id": "978dfd87-13b8-443a-892e-fca0380ef7d4",  
  "amount": 500,  
  "currency": "GBP",  
  "fee_refunded": 0,  
  "status": "REFUNDED",  
  "created": "2020-04-08T11:55:21",  
  "mode": "test"  
}
```

The ATLPay API is organized around REST. Our API has predictable, resource-oriented URLs, and uses HTTP response codes to indicate API errors. We use built-in HTTP features, like HTTP authentication and HTTP verbs, which are understood by off-the-shelf HTTP clients. We support cross-origin resource sharing, allowing you to interact securely with our API from a client-side web application (though you should never expose your secret API key in any public website's client-side code). JSON is returned by all API responses, including errors, although our API libraries convert responses to appropriate language-specific objects.

To make the API as explorable as possible, accounts have test mode and live mode API keys. There is no “switch” for changing between modes, just use the appropriate key to perform a live or test transaction. Requests made with test mode credentials never hit the banking networks and incur no cost.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`