Payment API Manual

About API

❑ POST API

Steps:
1. The customer sends their payment information to the merchant's web site.

2. The merchant web site posts the payment data to the Payment Gateway.

3. The Payment Gateway responds immediately with the results of the credit card transactions.

4. The merchant web site displays the appropriate message to the customer.

The communication method used to send messages to the Payment Gateway's server is the standard HTTP protocol over an SSL connection.

The communications with the cardholder (Steps 1 and 4) are developed completely by the merchant.

Step 1 should simply collect the payment data from the cardholder and
Step 4 should display the appropriate transaction receipt or declined
message.

In Step 2, transaction details should be delivered to the Payment Gateway using the POST method.

In Step 3, the transaction responses are returned in the body of the HTTP response in a JSON(**J**ava**S**cript **O**bject **N**otation) format.

For example:


```
{
    "result":{
        "id":"652385",
        "transaction":{
            "amount":1980,
            "transaction_id":"test_pgw_j4MJG3LN"
        },
        "time":"2014-09-28 22:35:33.566971",
        "status":"Approved",
        "reason":"Test PGW Approved (Normal Card)",
        "subs_id":"246993"
    }
}
```

Payment and Request Types

❑ one-time payment

- This is a payment method for a single service or a single product.
- If the same user makes a payment next times, they can use saved credit card information. The card information can be omitted from the payment.
- And in that case, you can request the last 4 digits of the user's card number in advance before payment.

❑ initial recurring payment

- This is a payment method for monthly subscription services and member-based subscription services.
- The billing cycle can be set flexibly in months, days.
- Automatically settles at regular intervals.
- Only the first payment will be made via API, and the second and subsequent payments will be made automatically.
- You can stop the recurring payment by sending a cancellation request.

❑ initial installment payment

- This is used for limited time membership services, or when you want to make people pay for large services or products in installments.
- Since there is no actual credit, it is in effect a recurring payment with a limited number of charges.
- The billing cycle can be set flexibly in months, days.
- Automatically settles at regular intervals.
- Only the first payment will be made via API, and the second and subsequent payments will be made automatically.

❑ refund request

- This can be used when you want to refund a past payment.
- Settlement within 20 days can be refunded.

❑ cancel request

- This is for recurring payment service.
- You can stop the recurring payment by sending a cancellation request.

Charge Types and Request Types

❑ Charge Type

- The charge type is one of the parameters required for API submission.
- The charge type differs depending on the Payment and Request Types.

Payment and Request Types	Charge Type
one-time payment :normal	6
one-time payment :saved credit card	6
initial recurring payment	1
initial installment payment	10
refund request	3
cancel request	5

❑ Request Types for one-time payment

- In the case of "one-time payment :omit card info", the content of the API parameter will differ depending on the request timing

Request Type for one-time payment		normal	saved credit card
A	Normal credit card transaction request (initial payment)	✔	✔
B	Saved credit card transaction request (2nd and subsequent payments)		✔
C	Saved credit card transaction request with updating credit card info (2nd and subsequent payments)		✔
D	Request last 4 digits saved card number (This is usually used to display on the user screen at a time before the B or C processes)		✔

API Request URL and Type

- **URL**

  Testing Server
  http://api2.stg.paymentapi.co:8081/payment2.php

  Live Server
  https://api2.paymentapi.co:8081/payment2.php

- **HTTP request type**

  POST

API Request Parameters (Input)

❑ Required and Optional

Parameters Name	Required: R / Optional: O							
Charge Type	1	3	5	6				10
Request Type	-	-	-	A	B	C	D	-
API Auth								
loginname	R							
charge_type	R							
site_id	R							
token	R							
API Command								
command	-	-	-	-	-	-	R	-
for Refund Request								
id	-	R	-	-	-	-	-	-
User Information								
email	O	-	-	O	O	O	-	O
subs_id	-	R	R	-	R	R	R	-
firstname	R	-	-	R	-	R	-	R
lastname	R	-	-	R	-	R	-	R
usrtel	O	-	-	O	-	O	-	O
zipcode	O	-	-	O	-	O	-	O
cardnumber	R	-	-	R	-	R	-	R
cardmonth	R	-	-	R	-	R	-	R
cardyear	R	-	-	R	-	R	-	R
cvv	R	-	-	R	-	R	-	R
user_ip	O	-	-	O	-	O	-	O
Currency								
currency	R	R	-	R	R	R	-	R
for Recurring and Installment Payment								
initial_amount	R	-	-	-	-	-	-	R
subsequent_amount	R	-	-	-	-	-	-	R
charging_term_unit	R	-	-	-	-	-	-	R
charging_term	R	-	-	-	-	-	-	R
charge_day	O	-	-	-	-	-	-	O
installments_number	-	-	-	-	-	-	-	R
next_charge_date	O	-	-	-	-	-	-	O
for One-time and Installment Payment (array)								
customer_id[0-n]	-	-	-	R	R	R	-	R
contents_id[0-n]	-	-	-	R	R	R	-	R
contents_name[0-n]	-	-	-	R	R	R	-	R
amount[0-n]	R	R	R	R	R	R	R	R

❑ Descriptions: API Auth

Parameters Name	Description
API Auth	
loginname	Login name assigned to merchant account. *ex) name123*
charge_type	Charge type **1**: initial recurring payment **3**: refund request **5**: cancel request **6**: One time payment **10**: initial installment payment
site_id	Site ID is an unique ID assigned to each site or store. *ex) 456*

token	Security key encrypted by MD5 algorithm.
	
	PHP example:
	`$loginName = 'name123';`
	`$pass = '#pass987!';`
	`$amount[0] = 1980;`
	`$amount[1] = 3000;`
	
	`$token = md5(sprintf('%s-.-%s-.-%s', $loginName, md5($pass), (string)array_sum($amount)));`
	`echo $token;`
	
	$token: Token
	$loginName: Login name assigned to merchant account.
	$pass: Password assigned to merchant account.
	$amount: Total amount to be charged.
	
	Token value($token) will be
	7cedb43ac7b9d85168804fa4f122216e

❑ Descriptions: API Command

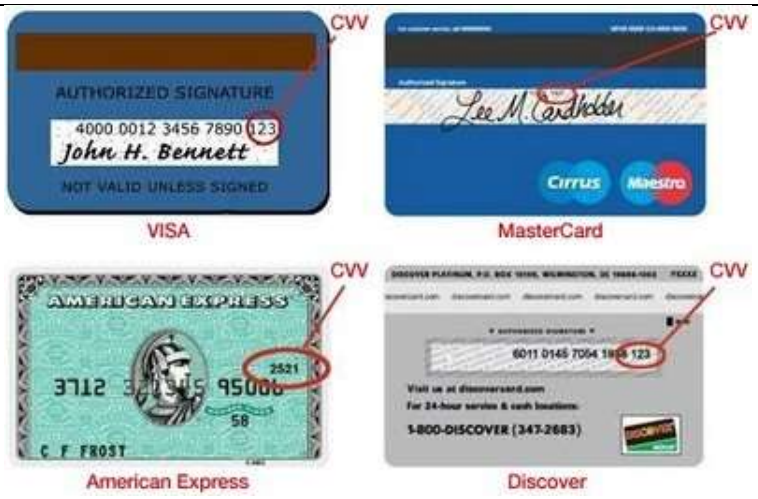
Parameters Name	Description
API Command	
command	**req_cn:** Request last 4 digits saved card number (13th digit or later)

❑ Descriptions: for Refund Request

Parameters Name	Description
for Refund Request	
id	Unique charge id **You can get this value from API Response.** *ex) 2987654*

❑ Descriptions: User Information

Parameters Name	Description
User Information	
email	(Optional) Card holder's email address **(max length 254)** *ex) name123@gmail.com*
subs_id	Card holder's subscriber id **You can get this value from API Response.** By using the **subs_id** stored in your database, you can make payments without credit card information. (request type **B**) If you send **both subs_id and credit card information,** you can update the credit card information associated with the target subs_id at the same time as the transaction. (request type **C**) For more information about **subs_id,** please refer to the chapter **Credit Card Transaction Response Variables.**
firstname	Card holder's first name *ex) john*
lastname	Card holder's last name *ex) Smith*
usrtel	(Optional) Card holder's telephone number **(only numeric)** *ex) 7175551234* *0355551234*
zipcode	(Optional) Card holder's zipcode **(only numeric)** *ex) 90210* *1001234*
cardnumber	Credit Card Number *ex) 4111222233334444* *378282246310005* *30569309025904*
cardmonth	2 digits Credit Card expiration Month *ex) 09*
cardyear	4 digits Credit Card expiration Year *ex) 2028*
cvv	The card security code. (3 digits or 4 digits)

	
user_ip	(Optional) Card user's IP address *ex) 255.255.255.255* *0.0.0.0*

❑ Descriptions: Currency

Parameters Name	Description
Currency	
currency	Transaction amount currency type. **USD:** US dollar **JPY:** Japanese yen

❑ Descriptions: for Recurring and Installment Payment

Parameters Name	Description
for Recurring and Installment Payment	
initial_amount	Initial amount to be charged *ex) 39.99* *3980*
subsequent_amount	Amount to be charged from next time *ex) 29.99* *2980*
charging_term_unit	Billing cycle unit **1**: Daily **2**: Monthly
charging_term	Number of days in the billing cycle (usually 30 days, but at least 7 days) Or specify the number of months (usually 1 month) *Daily → ex) 30* *90* *Monthly → ex) 1* *3*
charge_day	(Optional) billing reference date When the charging_term_unit is set to monthly, the charge day can be set. If not specified, the charge day will be based on the date of transaction. *ex) 31* *1*
installments_number	Payment frequency for installment billing *ex) 10* *5*
next_charge_date	(Optional) the next billing date If not specified, a date based on the billing cycle will be set. **If the charge_day is set, this field becomes required** and the date corresponding to the charge day and term must be entered. **(yyyy-mm-dd)** *ex) 2028-02-28* *2028-02-01*

❑ Descriptions: for One-time and Installment Payment

Parameters Name	Description
for One-time and Installment Payment (array)*	
customer_id[0-n]	Your service/product category id. If you don't have it, just put 1. **(only numeric and max length 9)** *ex) 987654321* *1*
contents_id[0-n]	Your service/product category id. If you don't have it, just put 1. **(only numeric and max length 9)** *ex) 123456789* *9*
contents_name[0-n]	Your service/product name. If you don't have it, it can be empty. *ex) service name 0001* *product qwerty*
amount[0-n]	Amount to be charged. *ex) 19.99* *980*

***How to use these parameters**

<1 item>

customer_id[0] → *ex) 987654321*
contents_id[0] → *ex) 123456789*
contents_name[0] → *ex) service name 0001*
amount[0] → *ex) 19.99*

<2 or more items>

customer_id[0] → *ex) 987654321*
contents_id[0] → *ex) 123456789*
contents_name[0] → *ex) service name 0001*
amount[0] → *ex) 19.99*

customer_id[1] → *ex) 987654322*
contents_id[1] → *ex) 123456790*
contents_name[1] → *ex) service name 0002*
amount[1] → *ex) 30.00*

customer_id[2] → *ex) 987654323*
contents_id[2] → *ex) 123456791*
contents_name[2] → *ex) service name 0003*
amount[2] → *ex) 7.50*
:

API Response Parameters (Output)

❑ JSON Layer

Parameters Name	JSON Layer		
	1	2	3
result	✔		
id		✔	
time		✔	
status		✔	
reason		✔	
subs_id		✔	
transaction		✔	
amount			✔
transaction_id			✔
command_response		✔	
req_cn			✔

For example:


```
{
    "result":{
        "id":"652385",
        "transaction":{
            "amount":1980,
            "transaction_id":"test_pgw_j4MJG3LN"
        },
        "time":"2014-09-28 22:35:33.566971",
        "status":"Approved",
        "reason":"Test PGW Approved (Normal Card)",
        "subs_id":"246993"
    }
}
```

❑ Response Parameters

Parameters Name	Response							
Charge Type	1	3	5	6				10
Request Type	-	-	-	A	B	C	D	-
result	✔	✔	✔	✔	✔	✔	✔	✔
id	✔			✔	✔	✔		✔
time	✔	✔	✔	✔	✔	✔	✔	✔
status	✔	✔	✔	✔	✔	✔	✔	✔
reason	✔	✔	✔	✔	✔	✔	✔	✔
subs_id	✔			✔	✔	✔		✔
transaction	✔			✔	✔	✔		✔
amount	✔			✔	✔	✔		✔
transaction_id	✔			✔	✔	✔		✔
command_response							✔	
req_cn							✔	

❑ Descriptions

Parameters Name	Descriptions
result	result group
id	Unique transaction identifier issued by Payment Gateway *ex) 652385*
time	The date and time that a Transaction became either 'Success' or 'Failed'. This is based on the **Pacific Time Zone (PT).** *ex) 2014-09-28 22:35:33.566971*
status	The status of a Transaction. **Approved, Refund requested, Cancel requested, Command requested** or **Failure.** *ex) Approved* *Cancel requested* *Failure*
reason	Text used to describe the Transaction Status. *ex) Test PGW Approved (Normal Card)*
subs_id	Subscriber id (user id) *ex) 246993*
transaction	transaction group
amount	Total amount to be charged. *ex) 19.99* *1980*
transaction_id	Unique transaction identifier issued by Payment Processor. Available only for Approved transaction. *ex) test_pgw_j4MJG3LN*
command_response	API command group
req_cn	last 4 digit saved credit card number (13th digit and later) *ex) 1111* *456* *98*