FlexShopper ™

Payment Method Documentation

FlexPrice Payment Platform

Patent Pending

Version 2015.06.09

Developer Prerequisites

To start your integration with the FlexPrice Payment Platform, you will need two accounts setup with FlexShopper, in addition to a software developer who is familiar with programming SDK/API integrations.

To use the FlexPrice Payment Platform, you must have:

• A Merchant account. A Merchant account enables you to invoice FlexShopper for the leased products, and collect payment from FlexShopper once your customer has transacted.

• A Sandbox account. The FlexPrice Sandbox provides an environment that simulates FlexPrice, in which you execute your Checkout integration without actually exchanging money. One of your test accounts represents you as the merchant, and the other test account represents a buyer. (Sandbox URL: sdk.sandbox.flexshopper.com)

• Programming experience. The integration involves adding a form and a button to your shopping cart or checkout page. When a user clicks this button, a POST is sent with your cart data (contained in hidden fields in the form) to the payment landing page at FlexShopper. The user will go through the payment process on the FlexShopper site, and when done, will be redirected back to your site with a URL that you provide. You will provide a few different URLs to handle different actions that may occur, such as a person being approved or denied for a FlexPrice Spending Limit. Once the user is redirected to one of the provided URLs, the last step is to call an API method to get the results of the customer's interaction with FlexPrice (usually an approval or denial).

Before You Start Coding

If you are not familiar with how APIs work, please contact us at: integration@flexshopper.com

INTEGRATION
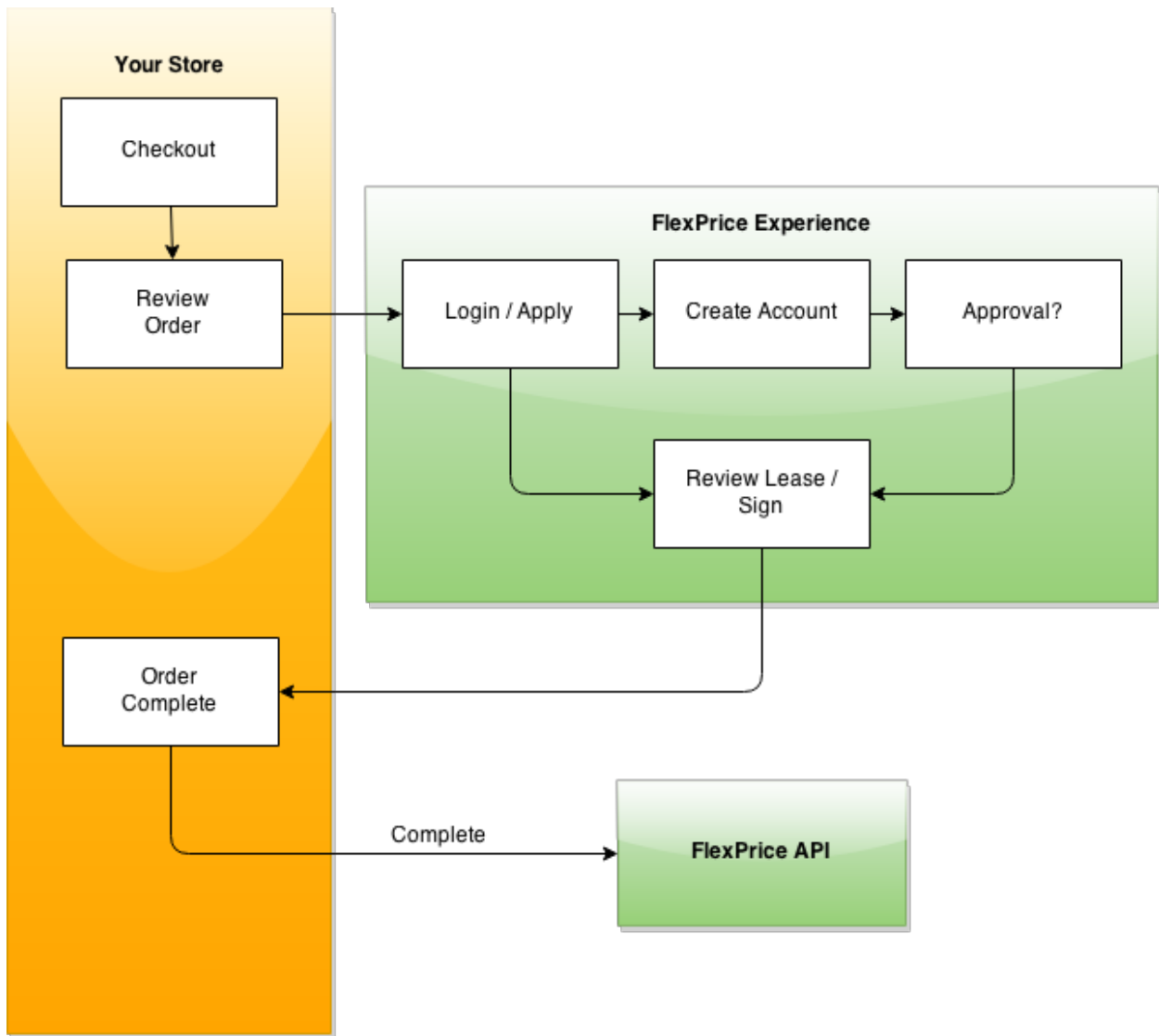
Note about tax and totals displayed on your checkout page: FlexShopper is tax exempt, so totals displayed on your checkout page do not include sales tax if a user selects "FlexShopper" as their payment option.

The Payment Method is composed of three parts:

1. The payment button

This is the button that's placed on the checkout page. It takes the user to the FlexShopper site to signup, create their agreement and finalize the transaction. Then, the user is redirected to the specified URL.

2. The transaction status API

Used to fetch the status of a transaction completed through the payment button

3. The weekly payment widget

Used to automatically estimate the weekly payment on your item page

The Payment Button

This is the list of required arguments for the payment button, along with a code example on how to implement it. All elements need to be passed inside a payload hash.

RetailerID

A 36 character UUID supplied at signup that uniquely identifies the retailer.

CallBackURL

A URL where the customer is sent back to after signing their agreement and completing the transaction.

NonSuccessfulURL

A URL where the customer is sent back to in the event of an error.

TransactionID

Up to a 256 character string which uniquely identifies the transaction for the retailer.

Items Items is a multi-dimensional array containing item information, each item takes following information:

1. item_sku

2. item_description (product name)

3. item_brand

4. item_cost

5. item_condition (values accepted are: new, refurbished, used)

6. qty

Special notes about the Items array:

• Support for bundles is available and can be sent like so:

• Individual item prices can be 0 so long as the order is net positive and above our 49.95 requirement

• Support for discounts is available and can be sent like so:

◦ item_sku starts with "DISCOUNT"

◦ item_cost is negative

◦ the overall order must still be a net positive and above our 49.95 USD requirement

DeliveryCost

The cost in USD to ship

DeliveryDate

The expected date the item will arrive in YYYY-MM-DD format

DeliveryMethod

Delivery method of the items to the customer:

1. parcel (Postal mail)

2. freight (Cargo Freight)

3. store (In-store pickup)

RequestSignature

A signature generated using JSON Web Token with the HMAC SHA-256 algorithm.

The payload for the encoding is a hash consistent of the following:

1. retailer_id (your Retailer ID)

2. cart_total (total item price + shipping)

The secret will be your Token

Example

```
<form action="http://sdk.flexshopper.com" method="post" id="paypal_payment_form"
target="_top">

<input type="hidden" name="payload[RetailerID]" value="{retailer ID}">

<input type="hidden" name="payload[CallBackURL]"
value="http://www.yourstore.com/callback/?order_id=228">

<input type="hidden" name="payload[NonSuccessfulURL]"
value="http://www.yourstore.com/cancel/?order_id=228">

<input type="hidden" name="payload[DeliveryCost]" value="25.00">

<input type="hidden" name="payload[DeliveryDate]" value="2014-10- 17">

<input type="hidden" name="payload[DeliveryMethod]" value="parcel">

<input type="hidden" name="payload[TransactionID]" value="228">

<input type="hidden" name="payload[Customer][first_name]" value="John">

<input type="hidden" name="payload[Customer][last_name]" value="Doe">

<input type="hidden" name="payload[Customer][street_addr1]" value="123 Any St">

<input type="hidden" name="payload[Customer][street_addr2]" value="Apt 456">

<input type="hidden" name="payload[Customer][city]" value="Delray Beach">

<input type="hidden" name="payload[Customer][state]" value="FL">

<input type="hidden" name="payload[Customer][zipcode]" value="33445">

<input type="hidden" name="payload[RequestSignature]"
value="eyJ0eXAOibGciOiJIUzI1NJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyZXRhaWxlcl9pZC
I6IjA2YmZhZTEyLTY0MzItMTFlMy1hZjE3LTc4MmJjYjExYjM1YiIsImNhcnRfdG90YWwi
OjIwMjF9.C1sMrx6cIg2gikLe0HOE-as-vYyzgvu7x-cU9thF7as">

<input type="hidden" name="payload[Items][0][item_description]" value="Xbox One">

<input type="hidden" name="payload[Items][0][item_sku]" value="MS-XBONE-2014">

<input type="hidden" name="payload[Items][0][item_cost]" value="499">

<input type="hidden" name="payload[Items][0][item_tax]" value="0">

<input type="hidden" name="payload[Items][0][item_brand]" value="Xbox One">

<input type="hidden" name="payload[Items][0][item_condition]" value="new">

<input type="hidden" name="payload[Items][0][qty]" value="2">

<input type="submit" value="Pay via FlexShopper">

</form>
```

The Transaction Status Endpoint

alias: GET Transactions

A read-only endpoint used to get information about your transactions.

Talking with the API

The endpoint is HTTP-Based and accepts POST requests with JSON in the body/response with the RetailerToken in the header for authentication.

Authentication

In order to make requests to the API, you must include your token in the header of the request as header "RetailerToken". The token should have been supplied to you at sign up.

Transaction Status Request:

POST /result HTTP/1.1

Host: http://sdk.flexshopper.com

RetailerToken: n9X9D75lOpJv49n

Content-Type: application/json

{

    "TransactionID": FPT123

}

Transaction Status – What's Returned?

A JSON body is returned with a status and details pertaining to it. Below are the possible statuses with examples of the returned JSON.

ok - the transaction was completed successfully as the customer signed a lease

{

    "status": "ok",

    "reference_id": "25CCC93FD93",

    "agreement_amount": 998,

    "shipping_amount": 8.32,

    "remaining": 1493.684CFC699CD7F4CFC699CD7F,

    "weekly_bill": 38.84,

    "TransactionID": FPT123,

    "items": {

        "3CFC699CD7F" : {

            "id": "3CFC699CD7F",

            "description": "Xbox One",

            "sku": "MS-XBONE-2014",

            "cost": 499,

            "shipping_cost": 4.16,

            "brand": "Xbox One",

            "condition": "new"

        },

        "4CFC699CD7F" : {

            "id": "4CFC699CD7F",

            "description": "Xbox One",

            "sku": "MS-XBONE-2014",

            "cost": 499,

            "shipping_cost": 4.16,

            "brand": "Xbox One",

            "condition": "new"

        }

    },

    "customer": {

        "first_name": "John",

        "last_name": "Doe",

```
                "email": "jdoe@gmail.com",

                "home_phone": "9993334444",

                "cell_phone": 3334445555",

                "street_addr1": "123 Test Dr.",

                "street addr2": "Apt. 2",

                "city": "Hollywood",

                "state": "FL",

                "zipcode": "33021"

        }

}
```

<mark>Note: Items with multiple quantities will appear as separate line items in the returned cart information.</mark>

declined - The customer's application has been denied

```
{

        "status": "declined",

        "customer": [ ...]

}
```

canceled - The customer has decided to cancel the transaction

```
{

        "status": "canceled",

        "customer": [ ...]

}
```

approved - The customer's application has been completed.

You may see this if the customer has been approved but has decided to cancel the transaction and go back to cart.

```
{

        "status": "approved",

        "remaining": 2000,

        "weekly_bill": 77.68,

        "customer": [ ...]

}
```

over_limit - The customer's spending limit is not high enough to purchase the

desired item(s)

```
{

        "status": "over_limit",

        "remaining": 200,

        "weekly_bill": 7.77,

        "customer": [ ...]

}
```

error - Something went wrong during the process

```
{

        "status": "error",

        "message": "invalid retailer_token" | "Invalid json payload" | ...

}
```

Generate Transaction ID Endpoint (optional)

The generate TransactionID endpoint is an ==optional== HTTP-based endpoint that accepts GET requests without any parameters. The generate TransactionID endpoint exists only for customers that prefer the FlexPrice payment processor to provide a unique transaction for them.

Authentication

In order to make requests to the API, you must include your token in the header of the request as header "RetailerToken". The token should have been supplied to you at sign up.

TransactionID Request:

GET /transaction-id HTTP/1.1

Host: http://sdk.flexshopper.com

RetailerToken: n9X9D75lOpJv49n

TransactionID – What's Returned?

A JSON body is returned with a single unique transaction ID.

```
{
    "TransactionID": FP987
}
```

GET Transactions

Get Transactions is an optional endpoint that allows a vendor to consume a batch of order data based on a date period.

Authentication

In order to make requests to the API, you must include your token in the header of the request as header "RetailerToken". The token should have been supplied to you at sign up.

Transactions Request:

GET /transaction[:/TransactionID] HTTP/1.1

Host: http://sdk.flexshopper.com

RetailerToken: n9X9D75lOpJv49n

TransactionID is optional. If TransactionID is passed, /transactions will return transaction information related to a specific transaction. Thus, it is essentially an alias of POST /result if an ID is provided.

Request Options:

/transaction[:/TransactionID]?[:from_date=<YYYY-MM-DD>][&:to_date=<YYYY-MM-DD>][&limit=<100>]

Transactions (single) Request:

/transaction/FPT123

Transactions (single) Response:


```
{
        "transaction_creation": "2015-05-12",
        "sign_date": null,
        "reference_id": "25CCC93FD93",
        "agreement_amount": 450,
        "shipping_amount": 4.16,
        "remaining": 2045.84,
        "weekly_bill": 22.84,
        "TransactionID": FPT123,
        "items": [
                {
                        "id": "3CFC699CD7F",
                        "description": "Xbox One",
                        "sku": "MS-XBONE-2014",
                        "cost": 225,
```

                        "shipping_cost": 2.08,

                        "brand": "Xbox One",

                        "condition": "new"

                },

                {

                        "id": "3CFC699CD7F",

                        "description": "Xbox One",

                        "sku": "MS-XBONE-2014",

                        "cost": 225,

                        "shipping_cost": 2.08,

                        "brand": "Xbox One",

                        "condition": "new"

                }

        ],

        "customer": {

                "first_name": "John",

                "last_name": "Doe",

                "email": "jdoe@gmail.com",

                "home_phone": "9993334444",

                "cell_phone": 3334445555",

                "street_addr1": "123 Test Dr.",

                "street addr2": "Apt. 2",

                "city": "Hollywood",

                "state": "FL",

                "zipcode": "33021"

        }

}

Transactions (multiple with parameters) Request:

/transaction?from_date=2015-05-01&limit=2

Transactions (multiple with parameters) Response:

{

        "FPT123": {

                "transaction_creation": "2015-05-12",

                "sign_date": null,

                "status": "over_limit",

                "agreement_amount": 718,

                "shipping_amount": 7.71,

                "remaining": 274.29,

                "weekly_bill": 38.84,

                "TransactionID": "FPT123",

                "items": [

                                {

                                                "id": "3CFC699CD7F",

                                                "description": "Xbox One",

                                                "sku": "MS-XBONE-2014",

                                                "cost": 499,

                                                "shipping_cost": 4.16,

                                                "brand": "Xbox",

                                                "condition": "new"

                                },

                                {

                                                "id": "3CF7779CD7F",

                                                "description": "Dyson DC40",

                                                "sku": "DY-DC40-2010",

                                                "cost": 219,

                                                "shipping_cost": 3.55,

                                                "brand": "Dyson",

                                                "condition": "new"

                                }

                ],

                "customer": [

                                "first_name": "John",

                                "last_name": "Doe",

                                "email": "jdoe@gmail.com",

                                "home_phone": "9993334444",

                                "cell_phone": 3334445555",

                                "street_addr1": "123 Test Dr.",

                                "street addr2": "Apt. 2",

                                "city": "Hollywood",

                                "state": "FL",

                                "zipcode": "33021"

                ]

        },

        "FPT124": {

                "transaction_creation": "2015-05-20",

                "sign_date": "2015-05-20",

```
"status": "ok",

"reference_id": "25CCC93FD93",

"agreement_amount": 449,

"shipping_amount": 4.16,

"remaining": 2046.84,

"weekly_bill": 22.84,

"TransactionID": FPT124,

"items": [

        {

                "id": "3CFC699CD7F",

                "description": "Xbox One",

                "sku": "MS-XBONE-2014",

                "cost": 449,

                "shipping_cost": 4.16,

                "brand": "Xbox",

                "condition": "new"

        }

],

"customer": {

        "first_name": "Jane",

        "last_name": "Doe",

        "email": "jane.doe@gmail.com",

        "home_phone": "5559994444",

        "cell_phone": 3337745555",

        "street_addr1": "987 Result Ln.",

        "street addr2": "",

        "city": "Hollywood",

        "state": "CA",

        "zipcode": "90048"

}

}

}
```

Shipment Tracking Information Endpoint

A read-only endpoint used to post information about the shipment of your orders to the customers.

Talking with the API

 The endpoint is HTTP-Based and accepts POST requests with JSON in the body/response with the RetailerToken in the header for authentication.

Authentication

In order to make requests to the endpoint, you must include your token in the header of the request as header "RetailerToken". The token should have been supplied to you at sign up.

Set Tracking Number

If you have a tracking number, you can use the /set-tracking-number endpoint, passing the following parameters:

• TransactionID – Your unique identifier for the transaction

• courier – Shipment provider code. Supported: ups, fedex, usps, dhl

• tracking_number – Shipment's Tracking number

Set Tracking Number Request

POST /set-tracking-number HTTP/1.1

Host: http://sdk.flexshopper.com

RetailerToken: n9X9D75lOpJv49n

Content-Type: application/json

```
 {

        "TransactionID": FPT123,

        "courier": "ups",

        "tracking_number": "123456Z789Z"

}
```

Set Tracking Number - What's returned?

 A JSON body is returned with a status and details pertaining to it. Below are the possible statuses with examples of the returned JSON.

ok - the transaction was completed successfully and the tracking number was stored

```
{

        "status": "ok",

        "message": "Tracking Number stored successfully"

        }
```

error - Something went wrong during the process

```
{
        "status": "error",
        "message": "invalid retailer_token" | "Invalid json payload" | ...
}
```

Set Delivery Date

If you only have a delivery date, you can use the /set-delivery-date endpoint, passing the following parameters:

• TransactionID – Your unique identifier for the transaction

• delivery_date – Delivery date in YYYY-MM-DD format

Set Delivery Date Request

 POST /set-delivery-date HTTP/1.1

Host: http://sdk.flexshopper.com

RetailerToken: n9X9D75lOpJv49n

Content-Type: application/json

```
{
        "TransactionID": FPT123,
        "delivery_date": "2014-09-31"
}
```

Set Tracking Number - What's returned?

A JSON body is returned with a status and details pertaining to it. Below are the possible statuses with examples of the returned JSON.

ok - the transaction was completed successfully and the delivery date was stored

```
{
        "status": "ok",
        "message": "Delivery Date stored successfully"
}
```

error - Something went wrong during the process

```
{
        "status": "error",
        "message": "invalid retailer_token" | "Invalid json payload" | …
        }
```

Order Manipulation API

Allows for modification of an order post-signature, because of the nature of the agreement between FlexShopper and the customer the allowed modifications are limited.

Order/Item Cancellation

For instances where an item/order has to be canceled.

- **TransactionID** – Your unique identifier for the transactions
- **sku_list** – an array of SKUs to cancel in the order (optional)

Set the Order/Item Cancellation Request – Canceling one item

POST /order-item-cancellation HTTP/1.1

Host: http://sdk.flexshopper.com

RetailerToken: n9X9D75lOpJv49n

Content-Type: application/json

{

"TransactionID": 123,

"sku_list": {"sku123" => 1}

}

Set the Order/Item Cancellation Request – Canceling the order

POST /order-item-cancellation HTTP/1.1

Host: http://sdk.flexshopper.com

RetailerToken: n646Y45lOpJv49n

Content-Type: application/json

{

"TransactionID": 123

}

The Order/Item Cancellation Response

ok – the order/item cancellation was successful

{

"status":"ok"

"message":"Order/Item successfully canceled"

"sku_list": {"sku123" => 1}

}

*note the sku_list element is only returned if it was provided as part of the initial request

Order/Item Return

For instances where an item/order has to be return.

- **TransactionID** – Your unique identifier for the transactions
- **sku_list** – an array of SKUs to cancel in the order (optional)

Set the Order/Item Return Request – Returning one item

POST /order-item-return HTTP/1.1

Host: http://sdk.flexshopper.com

RetailerToken: n9X9D75lOpJv49n

Content-Type: application/json

{

"TransactionID": 123,

"sku_list": {"sku123" => 1}

}

Set the Order/Item Return Request – Returning the order

POST /order-item- return HTTP/1.1

Host: http://sdk.flexshopper.com

RetailerToken: n646Y45lOpJv49n

Content-Type: application/json

{

"TransactionID": 123

}

The Order/Item Return Response

ok – the order/item return was successful

{

"status":"ok"

"message":"Order/Item successfully returned"

"sku_list": {"sku123" => 1}

}

*note the sku_list element is only returned if it was provided as part of the initial request