



Web Integration Direct Post One Off CC

Integration Guide Version 1.0
27/04/2016

Table of Contents

Overview	1
Introduction	1
Checking that Web Payments is enabled for your account	2
Direct Post Payments Workflow	2
Overview	2
Registering the Payment	3
Authentication	3
Register Payment Request Data	4
Bad Request Response	12
Forbidden Response	12
Successful Request Response	12
Processing the Payment	14
POST Data	14
Retrieving the Result	16
Successful Request Response	16
Payment Resource	16
Processing Result Status	19
Validation Error (2)	19
Processing Error - Please Retry (3)	19
Processing Error - Please Check (4)	19
Payment Processed (5)	19
Appendices	25
Appendix A - List of responses	25
Appendix B - List of error codes and messages	27

Overview

Introduction

Direct post payments enable merchants to accept payments through their own website or e-commerce platform by registering a payment request before posting payment data directly to Flo2Cash for processing.

Before you use the direct post web integration for making payments, please check that your account is subscribed to the web payments channel.

Checking that Web Payments is enabled for your account

Before using the direct post web integration for making payments you first need to ensure that Web Payments is enabled for your account. To do this, log in to your Flo2Cash Merchant Console available at <https://secure.flo2cash.co.nz/client/> supplying the username and password that you received from Flo2Cash when creating your account.

Once you have successfully logged in to the Merchant Console, navigate to the “Channel Settings” section via the menu available on the left hand side of the screen. The Channel Settings page shows the Flo2Cash channels you currently subscribe; if subscribed Web Payments will appear in the list.

IMPORTANT

Flo2Cash reserve the right to change the schema of both requests and responses used in this product but will only do so where additional elements or data are providing. Nothing that exists will be removed or changed. It is the integrators responsibility to ensure their clients do not break when new elements are available.

Direct Post Payments Workflow

Overview

Making a payment using direct post involves three steps:

1. Registering the payment request
2. Posting payment details
3. Retrieving the result

The following steps provide some context using an example where a customer is purchasing a product from a merchant’s web site.

1. The customer visits the merchant’s website and selects their purchase
2. The merchant’s website registers a payment request through an API call to Flo2Cash
3. A payment URL is returned to the merchant’s website
4. A payment form is rendered to the customer allowing them to select the payment method (if more than one is offered by the merchant) and enter any associated details such as card data. The form action is set to post directly to the URL returned in step 3
5. The Customer selects the preferred payment method and in the case of a card payment enters their card details before submitting the form
6. The form data including card details are posted directly to the payment URL provided in step 3
7. If the data is valid a payment will be processed
8. If the Merchant Notification Service is being used a notification will be sent (asynchronously)
9. The customer is returned to the return URL provided in step 2 which contains the unique identifier for the payment
10. The merchant requests the payment result and displays this to the customer

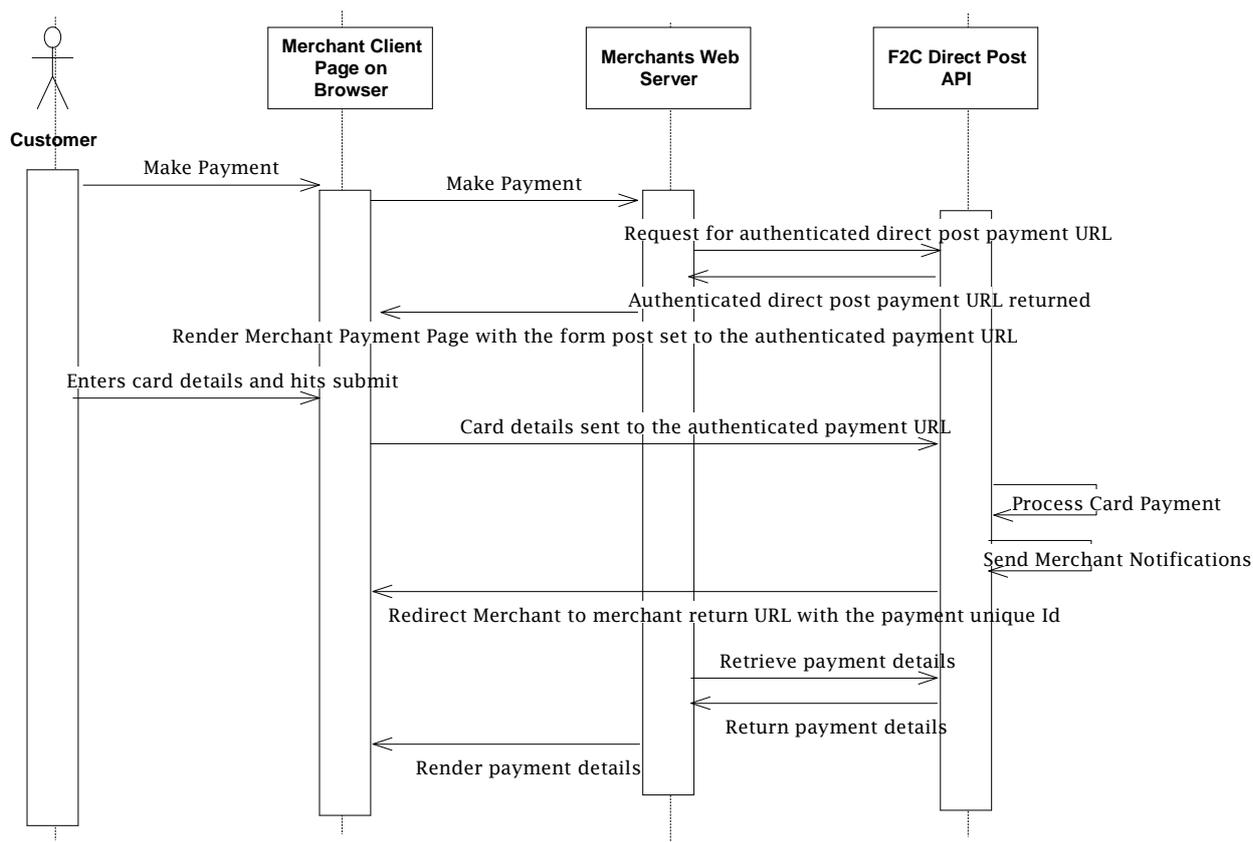


Figure 1 - Direct Post Payments Overview

Registering the Payment

The first step to making a payment via direct post is to register the payment request; if successful this will result in unique payment identifier and payment URL being returned. To register a payment, request a HTTP POST should be made to the following URI:

<https://secure.flo2cash.co.nz/integration/payments/dp>

Authentication

Each request to the API is authenticated and the post should include a basic authorization header with your API key included - this can be obtained from the Merchant Console at *Channel Settings > Web Payments > Direct Post Settings*.

Please ensure that you have an “Authorization” HTTP header in your request like the below example:

Authorization: Basic QWxhZGRpbjpvYVUHNlc2FtZQ==

Please refer here http://en.wikipedia.org/wiki/Basic_access_authentication for more information on Basic Access Authentication for HTTP.

If the credentials are not provided, corrupt or invalid a **401 - Unauthorised** response will be returned along with a *WWW-Authenticate* header.

Register Payment Request Data

Requests can be made in either XML or JSON by setting the Content-Type header to match the format provided.

Content-Type: application/xml

Content-Type: application/json

Response can be sent in either XML or JSON by setting the Accept header to match the format provided.

Accept: application/xml

Accept: application/json

The table below indicates all of the data that can be provided when registering a payment request. Some data are mandatory whilst some are optional but if provided will be stored with the subsequent payment.

Element	Child Element	Description	Required/Optional	Type	Min-Max	Comments
payment		Payment Element	R			
	subAccount	Sub account through which the payment will be processed	R	String		Must exist under merchants account
	requestDuration	The number of minutes the payment request is active before expiring	O	Integer	0 - n	If not provided this defaults to 20 minutes. A value of 0 indicates the request does not expire.
	amount	Amount the payment will be charged for	R	Decimal	0.01 - n	Must be greater than 0
	currency	Currency of payment	R	String	3	ISO currency code - NZD is the only valid value at present
	returnUrl	The URL the customer will be returned to	R	String	11 - 1024	Valid URL beginning http:// or https://
	notificationUrl	The URL MNS notifications will be sent to	O	String	11 - 1024	Valid URL beginning http:// or https://

	reference	Merchant defined reference for the payment	O	String	0 – 50	
	particulars	Merchant defined particulars for the payment	O	String	0 – 50	
	receiptEmail	Email for the receipt to be sent to	O	String	0 – 256	Must be a valid email address. If provided a PDF receipt will be sent to this address
	deviceId	Unique identifier of the customer's device	O	String	0 - 50	
	deviceDescription	Description of the customer's device	O	String	0 - 255	
	longitude	Longitude of customer's geo-location	O	String	0 – 8	
	latitude	Latitude of customer's geo-location	O	String	0 – 8	
payment.customer		Customer Element	O			
	companyName	Name of the company	O	String	0 - 50	
	title	Salutation title	O	String	0 – 50	
	firstName	First name	O	String	0 – 128	
	lastName	Last name	O		0 – 128	
	dateOfBirth	Date of birth	O	String	8	Formatted YYYYMMDD
	homePhone	Home phone	O	String	0 – 50	
	workPhone	Work phone	O	String	0 – 50	
	mobilePhone	Mobile phone	O	String	0 – 50	
	fax	Fax number	O	String	0 – 50	
	email	Email address	O	String	0 – 256	
	addressLine1	First line of address	O	String	0 – 128	
	addressLine2	Second line of address	O	String	0 – 128	
	addressLine3	Third line of address	O	String	0 – 128	
	addressSuburb	Suburb of address	O	String	0 – 50	

	addressCity	City of address	0	String	0 – 50	
	addressCountry	Country of address	0	String	2	ISO Alpha 2 Country Code http://en.wikipedia.org/wiki/ISO_31661_alpha-2
	addressPostcode	Postcode of address	0	String	0 – 50	
payment.shippingAddress		ShippingAddress Element	0			
	line1	First line of address	0	String	0 – 128	
	line2	Second line of address	0	String	0 – 128	
	line3	Third line of address	0	String	0 – 128	
	suburb	Suburb of address	0	String	0 – 50	
	city	City of address	0	String	0 – 50	
	country	Country of address	0	String	2	ISO Alpha 2 Country Code http://en.wikipedia.org/wiki/ISO_31661_alpha-2
	postcode	Postcode of address	0	String	0 – 50	
payment.shoppingCart		ShoppingCart Element	0			
	name	Name of the cart	0	String	0 – 50	
	version	Version of the cart	0	String	0 – 50	
	description	Description of the cart	0	String	0 – 50	
payment.shoppingCart.items		Item Element	0			
payment.shoppingCart.items.cartItem		CartItem Element	0			Can be more than one
	description		0	String	0 – 50	
	code		0	String	0 – 50	
	quantity		0	Number		
	unitCost		0	Decimal		
payment.customData		CustomData Element	0			
payment.customData.data		Data Element	0			Can have more than one

	name	Merchant defined name	R		1 – 50	
	value	Merchant defined value	R		0 – 255	

XML Example

```

<payment>
  <subAccount>10002</subAccount>
  <requestDuration>300</requestDuration>
  <amount>10</amount>
  <currency>NZD</currency>
  <returnUrl>http://www.example.com/return</returnUrl>

  <notificationUrl>http://www.example.org/notification</notificationUrl>
  <reference>Ref</reference>
  <particulars>Par</particulars>
  <receiptEmail>test@example.org</receiptEmail>
  <deviceId>id</deviceId>
  <deviceDescription>Samsung Galaxy s6</deviceDescription>
  <longitude>-36.749767</longitude>

  <latitude>174.7010251</latitude>
  <customer>
    <companyName>Company</companyName>
    <title>Mr</title>
    <firstName>First</firstName>
    <lastName>Last</lastName>
    <dateOfBirth>1988-01-02T00:00:00</dateOfBirth>
    <homePhone>HP</homePhone>
    <workPhone>WP</workPhone>
    <mobilePhone>MP</mobilePhone>
    <fax>F</fax>
    <email>test@example.org</email>
    <addressLine1>Line 1</addressLine1>
    <addressLine2>Line 2</addressLine2>
    <addressLine3>Line 3</addressLine3>
    <addressSuburb>BSuburb</addressSuburb>
    <addressCity>Auckland</addressCity>
    <addressCountry>NZ</addressCountry>
    <addressPostCode>0614</addressPostCode>
  </customer>

  <shippingAddress>
    <line1>SLine1</line1>
    <line2>SLine2</line2>
    <line3>SLine3</line3>
    <suburb>SSuburb</suburb>
    <city>SCity</city>
    <country>NZ</country>
  <postCode>0614</postCode>
</shippingAddress>

```

```
<shoppingCart>  
  <name>CartName</name>
```

```
<version>1</version>
<description>Cart</description>
  <items>
    <cartItem>
      <description>Des1</description>
      <code>code1</code>
      <quantity>1</quantity>
      <unitCost>1</unitCost>
    </cartItem>

    <cartItem>
      <description>Des2</description>
      <code>code2</code>
      <quantity>2</quantity>
      <unitCost>2</unitCost>
    </cartItem>

    <cartItem>
      <description>Des3</description>
      <code>code3</code>
      <quantity>3</quantity>
      <unitCost>3</unitCost>
    </cartItem>
  </items>

</shoppingCart>

<customData>
  <data>
    <name>N1</name>
    <value>V1</value>
  </data>
  <data>
    <name>N2</name>
    <value>V2</value>
  </data>
  <data>
    <name>N3</name>
    <value>V3</value>
  </data>
</customData>

</payment>
```

JSON Example

```
{
  payment:{
    subAccount:10002,
requestDuration:300,
amount:10,
currency:'NZD',
    returnUrl:'http://www.example.com/return',

notificationUrl:'http://www.example.org/notification',
reference:'Ref',      particulars:'Par',
    receiptEmail:'test@example.org'
```

```
    deviceId:'id',
deviceDescription:'De',
longitude:-36.749767,
latitude:174.7010251,
customer:{
    companyName:'Company',
title:'Mr',
firstName:'First',
lastName:'Last',
    dateOfBirth:'1988-01-
02T00:00:00',      homePhone:'HP',
workPhone:'WP',
mobilePhone:'MP',      fax:'F',
    email:'test@example.org',
addressLine1:'Line 1',
addressLine2:'Line 2',
addressLine3:'Line 3',
addressSuburb:'BSuburb',
addressCity:'Auckland',
addressCountry:'NZ',
addressPostCode:396
    },
    shippingAddress:{
line1:'SLine1',
line2:'SLine2',
line3:'SLine3',
suburb:'SSuburb',
city:'SCity',
country:'NZ',
postCode:396
    },
    shoppingCart:{
name:'CartName',
version:1,
description:'Cart',
items:{
cartItem:[
{
description:'Des1',
code:'code1',
```

```

quantity:1,
unitCost:1

},
{

description:'Des2',
code:'code2',
quantity:2,
unitCost:2

},
{

description:'Des3',
code:'code3',
quantity:3,
unitCost:3

}

]

},
customData:{
data:[
{
name:'N1',
value:'V1'
},
{
name:'N2',
value:'V2'
},
{
name:'N3',
value:'V3'
}
]
}
}
}

```

If the request results in an error the body of the response will contain an entity with the following structure - specific error examples are found below.

Element	Child Element	Description	Type	Comments
error		Error Element		
	type	Type of error	String	
	httpStatusCode		String	
error.messages		Messages Element		
	message	Description of the error	String	Can be more than one

Bad Request Response

If the data provided is not valid in accordance with the rules above, then a **400 - Bad Request** response will be returned with a list of error messages provided in the body of the response.

XML Example

```
<error>
  <type>BAD REQUEST</type>
  <statusCode>400</statusCode>
  <messages>
    <message>Amount to charge must be greater than 0</message>
  </messages>
</error>
```

JSON Example

```
{
  error:{
    type:'BAD REQUEST',    statusCode:400,    messages:{
      message:'Amount to charge must be greater than 0'
    }
  }
}
```

Forbidden Response

If the merchant is not subscribed to the web payments payment gateway channel, then a **403 - Forbidden** response will be returned with a message in the body of the response "Web payments are not enabled for this account"

XML Example

```
<error>
  <type>FORBIDDEN</type>
  <statusCode>403</statusCode>
  <messages>
    <message>Web payments are not enabled for this account</message>
  </messages>
</error>
```

JSON Example

```
{
  error:{
    type:'FORBIDDEN',
    statusCode:403,
    messages:{
      message:'Web payments are not enabled for this account'
    }
  }
}
```

Successful Request Response

If the data provided is valid then a new request will be generated and a **201 - Created** response will be returned containing the authenticated direct post payment URL, the

unique identifier for the payment and the expiry date/time when the payment URL becomes inactive.

XML Example

```
<paymentRequest>
  <id>57F31EFE-01AE-4D80-8EB5-7A16D5979DB5</id>
  <url>https://secure.flo2cash.co.nz/integration/payments/dp/57F31EFE-01AE-4D80-8EB5-7A16D5979DB5</url>
  <expiryDate>2015-10-07T15:03:20.5675691+13:00</expiryDate>
  <chargeConvenienceFee>true</chargeConvenienceFee>
  <convenienceFees>
    <convenienceFeeType>Percentage</convenienceFeeType>
    <convenienceFeeValue>5</convenienceFeeValue>
    <convenienceFeeAmount>5.10</convenienceFeeAmount>
  </convenienceFees>
</paymentRequest>
```

JSON Example

```
{
  id: "57F31EFE-01AE-4D80-8EB5-7A16D5979DB5"
  url: "'https://secure.flo2cash.co.nz/integration/payments/dp/57F31EFE-01AE-4D80-8EB5-7A16D5979DB5'"
  expiryDate: "2015-10-14T08:31:57.9580779+13:00"
  chargeConvenienceFee: true    convenienceFees: {
    convenienceFeeType: "Percentage"
  }
  convenienceFeeValue: 5
  convenienceFeeAmount: 5.10
}
```

Element	Child Element	Description
paymentRequest		Details to take a payment via direct post
	id	Unique id for a direct post payment
	url	Unique url where card details need to be posted
	expiryDate	Expiry date of the URL, posting card details after the expiry date will not process any payments
	chargeConvenienceFees	A flag which indicates whether convenience fees will be charged the direct post payment. If this is true, then the total amount charged to customers will be the amount passed in the request + the convenience fee value set in the Merchant Console
paymentRequest.ConvenienceFees	convenienceFeeType	One of the three values: <ol style="list-style-type: none"> 1. None 2. Percentage 3. Amount (Flat fees per payment)

	convenienceFeeValue	Value of the convenience fees to be charged for a direct post payment. If the convenience fee type is percentage then the convenience fee value will be in percentage, if the type is amount then the value will be monetary
	convenienceFeeAmount	Amount of the convenience fees to be charged for the current transaction.

Processing the Payment

Once a payment request has been registered and a payment URL returned, the payment form can be rendered to the customer.

IMPORTANT

To avoid card data touching your server and increasing your PCI-DSS scope you must ensure the action URL of the HTML payment form is set to the URL returned from the API request described above. This will ensure the form data is sent direct to Flo2Cash from the customer's browser and not via your servers.

IMPORTANT

Merchants should always host their payment form on an SSL enabled website to provide confidence to their customers that the solution is secure.

POST Data

The name attributes of the HTML form elements need to be properly defined; here is a list of expected form elements when the page is posted.

HTML Form Field Name	Description	Required /Optional	Type	Min-Max	Comments
PaymentMethod	Method of payment gateway to use. 1. Standard for Visa/MasterCard/AMEX/Diners	R	String	8	Valid values are • standard
CardNumber	Full number (PAN) of card to charge	R	String	20	<ul style="list-style-type: none"> • Digits only • Passes Luhn check • Accepted card scheme

CardCSC	Card Security Code found on the back of the card	R	String	3-4	VISA/MasterCard/Diners will have 3 digit CSC American Express will have 4 digit CSC
CardExpiryMonth	Month the card expires in	R	String	2	Should be a valid month
CardExpiryYear	Year the card expires in	R	String	2	Valid year
NameOnCard	Name of the card holder as it appears on the card	R	String	1-128	

HTML Example

```

<html>
  <head></head>

  <body>

    <form
action="https://secure.flo2cash.co.nz/integration/payments/dp/b2097104d57c-
4539-847b-d1f75ffc8561" method="POST">
      Card Number <input type="text"
name="CardNumber"
value="4987654321098769"><br>
      Card Name <input type="text" name="NameOnCard" value="A Test"><br>
      Card CSC <input type="text" name="CardCSC" value="555"><br>
      Card Expiry Month <input type="text" name="CardExpiryMonth"
value="05"><br>
      Card Expiry Year
<input type="text" name="CardExpiryYear" value="17"><br>
      Payment
Method <input type="text" name="PaymentMethod"
value="standard"><br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>

```

Once the payment is processed, the user will be redirected to the return URL with the payment id appended as a parameter.

E.g. If the return URL had a value of <http://www.example.org> then the user will be redirected to <http://www.example.org?id=24640497-C1A6-4724-874B-6B744B641579>

If the request had been used previously then in the return URL a message parameter will be appended stating that "Payment request has already been used"

E.g. <http://www.example.com/PaymentUniqueId=e6f634d8-c856-4f52-a65ed3e4b344f4ae&Message=Payment request has already been used>

E.g. <http://www.example.com/PaymentUniqueId=e6f634d8-c856-4f52-a65ed3e4b344f4ae&Message=Payment request has expired>

Retrieving the Result

Once the customer has been returned the merchant needs to retrieve the outcome of the payment to inform their customer of the result; there are four possible results:

1. Validation failed
2. Processing error - can be retried
3. Processing error - check Merchant Console
4. Payment processed

The user needs to make a HTTP GET request to

<https://secure.flo2cash.co.nz/integration/payments/57F31EFE-01AE-4D80-8EB57A16D5979DB5>

Where **57F31EFE-01AE-4D80-8EB5-7A16D5979DB5** will be the payment id found on the return URL and also returned when registering the payment request.

Each request to the API is authenticated and so the request should include a basic authentication header with your API key included.

Please ensure that you have an "Authorization" header in your initial request. Here is an example of the authorization header in the request:

Authorization: Basic QWxhZGRpbjpvGVuIHNlc2FtZQ==

Please refer here http://en.wikipedia.org/wiki/Basic_access_authentication for more information on Basic Access Authentication.

If the credentials are not provided, corrupt or invalid a **401 - Unauthorised** response will be returned along with a *WWW-Authenticate* header.

Successful Request Response

If the id is found, then then a **200 - OK** response will be returned containing the payment resource.

Payment Resource

The resource fields that get returned are explained below.

Element	Child Element	Description	Type
payment		payment Element	
	id	Unique Identifier to process the payment	Guid
	url	Unique Url to set the merchant's Form Post to	String
	expiryDate	Date/Time of when the unique payment Url expires	DateTime
	requestDuration	The number of minutes the payment request is active before expiring	Integer
	merchant	Merchant Id for the request	String
	subAccount	Sub account through which the payment will be processed	String
	amount	Amount the payment will be charged for	Decimal
	currency	Currency of payment	String
	returnUrl	The URL the customer will be returned to	String
	notificationUrl	The URL MNS notifications will be sent to	String

	reference	Merchant defined reference for the payment	String
	particulars	Merchant defined particulars for the payment	String
	receiptEmail	Email for the receipt to be sent to	String
	deviceId	Unique identifier of the customer's device	String
	deviceDescription	Description of the customer's device	String
	longitude	Longitude of customer's geo-location	String
	latitude	Latitude of customer's geo-location	String
payment.customer		Customer Element	
	companyName	Name of the company	String
	title	Salutation title	String
	firstName	First name	String
	lastName	Last name	
	dateOfBirth	Date of birth	String
	homePhone	Home phone	String
	workPhone	Work phone	String
	mobilePhone	Mobile phone	String
	fax	Fax Number	String
	email	Email Address	String
	addressLine1	First line of address	String
	addressLine2	Second line of address	String
	addressLine3	Third line of address	String
	addressSuburb	Suburb of address	String
	addressCity	City of address	String
	addressCountry	Country of address	String
	addressPostcode	Postcode of address	String
payment.shippingAddress		ShippingAddress Element	
	line1	First line of address	String
	line2	Second line of address	String
	line3	Third line of address	String
	suburb	Suburb of address	String
	city	City of address	String
	country	Country of address	String
	postCode	Postcode of address	String
payment.shoppingCart		ShoppingCart Element	
	name	Name of the cart	String
	version	Version of the cart	String

	description	Description of the cart	String
payment.shoppingCart.items		Item Element	
payment.shoppingCart.items.cartItem		CartItem Element	
	description		String
	code		String
	quantity		Number
	unitCost		Decimal
payment.customData		CustomData Element	
payment.customData.data		Data Element	
	name	Merchant defined name	
	value	Merchant defined value	
payment.processingResult			
	status	Status for the request, please refer to appendix B	Integer
	errorCode	A unique error code indicating the error that has occurred	String
	errorMessage	A text description of the error code	String
	paymentMethod	Payment method used to process the payment	String
	transactionNumber	Transaction number	String
	transactionStatus	Status for the transaction	String
	responseCode	Response code for the payment, please refer to appendix A	Integer
	responseMessage	Response message	String
Payment.convenienceFees			
	convenienceFeeType	One of these three values: 1. None 2. Percentage 3. Amount (Flat fees per payment)	String
	convenienceFeeValue	Percentage or flat amount of convenience fees per payment	Decimal
	convenienceFeeAmount	Amount of convenience fees for this payment	Decimal

Processing Result Status

There are four statuses a payment result can be in. The table below details the four possible statuses.

Status	Description
2	Validation error
3	Processing error – please retry
4	Processing error – please check before retrying
5	Request was processed

Validation Error (2)

If the data provided by the merchant in the second step (HTTP POST from merchant's website) is not valid against the rules defined above the payment will not be processed and the result status will be set to 2 with an error code and message set in the **errorCode** and **errorMessage** fields respectively. The possible validation error codes are available in Appendix B.

Processing Error - Please Retry (3)

If the data is valid then a transaction will be processed, however, if for any reason an error happens meaning the transaction does not process then the result status will be 3; indicating to the merchant that something unexpected has gone wrong but the transaction did not happen. If this result is encountered, then the merchant can retry the payment.

Note: To retry the payment the merchant should start over by making a new request for payment. The failed payment request cannot be used again.

Processing Error - Please Check (4)

If the data is valid then a transaction will be processed, however, if for any reason an error happens then the result status will be 4; indicating to the merchant that something unexpected has gone wrong. If this result is encountered, then the merchant should first check through their merchant console that a transaction has not taken place and if not they can retry the payment.

Note: To retry the payment the merchant should start over by making a new request for payment. The failed payment request cannot be used again.

Payment Processed (5)

If the data is valid then a transaction will be processed and the result will be available in the following fields:

A unique transaction number will be found in the **transactionNumber** element

The transaction status will be found in the **transactionStatus** element; the possible transaction states are:

Status	Description	Comments
2	Successful	The transaction was approved and successful
3	Failed	The transaction failed to process

4	Blocked	The transaction was blocked from processing due to transaction limits on the merchant's account
11	Declined	The transaction has been declined
0	Unknown	Usually due to an upstream error whereby the result has not been provided for the transaction submitted – Flo2Cash will generally update these automatically when the result becomes available

Response code will be found in the **responseCode** element - see appendix A
Response message will be found in the **responseMessage** element - see appendix A
The method for payment will be found in the **paymentMethod** element

XML Example

```

<payment>
  <merchant>20002</merchant>
  <subAccount>10002</subAccount>
  <id>c750a443-abb6-412b-a908-8997959f517f</id>

  <url>https://secure.flo2cash.co.nz/integration/payments/dp/c750a443abb6-412b-a908-8997959f517f</url>
  <requestDuration>300</requestDuration>
  <expiryDate>2015-05-21T01:29:43</expiryDate>
  <amount>10.00</amount>
  <currency>NZD</currency>
  <returnUrl>https://www.example.org/return.php</returnUrl>
  <notificationUrl>https://www.example.org/noti.php</notificationUrl>
  <reference>Ref</reference>
  <particulars>Par</particulars>
  <receiptEmail>test@example.org</receiptEmail>
  <deviceId />
  <deviceDescription />
  <longitude />
  <latitude />

  <customer>
    <companyName>N1</companyName>
    <title>T</title>
    <firstName>FN</firstName>
    <lastName>LN</lastName>
    <dateOfBirth>1988-01-02T00:00:00</dateOfBirth>
    <homePhone>HP</homePhone>
    <workPhone>WP</workPhone>
  <mobilePhone>MP</mobilePhone>
    <email>S</email>
    <fax>F</fax>
    <addressLine1>CLine1</addressLine1>
    <addressLine2>CLine2</addressLine2>
    <addressLine3>CLine3</addressLine3>
    <addressSuburb>CSuburb</addressSuburb>
    <addressCity>CCity</addressCity>
  </customer>
</payment>

```

```
    <addressCountry>NZ</addressCountry>
    <addressPostCode>0614</addressPostCode>
</customer>

<shippingAddress>
  <line1>SLine1</line1>
  <line2>SLine2</line2>
  <line3>SLine3</line3>
  <suburb>SSuburb</suburb>
  <city>SCity</city>
  <country>NZ</country>
  <postCode>0614</postCode>
</shippingAddress>

<shoppingCart>
  <name>CartName</name>
  <version>1</version>
<description>Cart</description>
  <items>
    <cartItem>
<description>Des1</description>
      <code>code1</code>
      <quantity>1</quantity>
```

```
      <unitCost>1.00</unitCost>
    </cartItem>
    <cartItem>
      <description>Des2</description>
      <code>code2</code>
      <quantity>2</quantity>
      <unitCost>2.00</unitCost>
    </cartItem>
    <cartItem>
      <description>Des3</description>
      <code>code3</code>
      <quantity>3</quantity>
      <unitCost>3.00</unitCost>
    </cartItem>
  </items>
</shoppingCart>

<customData>
  <data>
    <name>N1</name>
    <value>V1</value>
  </data>
  <data>
    <Name>N2</name>
    <Value>V2</value>
  </data>
  <data>
    <Name>N3</name>
    <Value>V3</value>
  </data>
```

```
</customData>

<processingResult>standard
  <errorCode></errorCode>
  <errorMessage />
  <paymentMethod>standard</paymentMethod>
  <transactionNumber>P150500000050205</transactionNumber>
  <transactionStatus>Successful</transactionStatus>
  <responseCode>0</responseCode>
  <responseMessage>Successful</responseMessage>
</processingResult>
<convenienceFees>
  <convenienceFeeType>Amount</convenienceFeeType>
  <convenienceFeeValue>0.1500</convenienceFeeValue>
  <convenienceFeeAmount>0.15</convenienceFeeAmount>
</convenienceFees>
</payment>
```

JSON EXAMPLE

```
{
  payment:{
    id:'c750a443-abb6-412b-a908-8997959f517f',
    url:'https://secure.flo2cash.co.nz/integration/payments/dp/c750a443-abb6-412b-a908-8997959f517f',
    expiryDate:'2015-05-21T01:29:43',
    requestDuration:300,
    merchant:20002,
    subAccount:10002,          amount:10,
    currency:'NZD',
    returnUrl:'http://www.example.org/return.php',
    notificationUrl:'http://www.example.org/noti.org',
    reference:'Ref',          particulars:'Par',

    receiptEmail:'test@example.org'
    deviceId:""
    deviceDescription:""
    longitude:""          latitude:""
    customer:{
      companyName:'N1',
      title:'T',
      firstName:'FN',
      lastName:'LN',
      dateOfBirth:'1988-01-02T00:00:00',
      homePhone:'HP',          workPhone:'WP',
      mobilePhone:'MP',          email:'S',
      fax:'F',
      addressLine1:'CLine1',
      addressLine2:'CLine2',
      addressLine3:'CLine3',
      addressSuburb:'CSuburb',
      addressCity:'CCity',
      addressCountry:'NZ',
      addressPostCode:396
    },
    shippingAddress:{
      line1:'SLine1',
      line2:'SLine2',
      line3:'SLine3',
      suburb:'SSuburb',
      city:'SCity',
      country:'NZ',
      postcode:396
    },
    shoppingCart:{
      name:'CartName',
      version:1,
      description:'Cart',
      items:{
        cartItem:[
          {
            description:'Des1',
            code:'code1',
            quantity:1,
            unitCost:1
          }
        ]
      }
    }
  }
}
```


Appendices

Appendix A - List of responses

Status	Code	Message	Description
Success	0	Successful	Transaction has completed successfully
Declined	200	Declined - Insufficient funds	Transaction declined due to insufficient funds
Declined	201	Declined - Expired card	Transaction declined due to expired card
Declined	202	Declined - Bank declined	Transaction declined by the bank with no additional reason
Declined	203	Declined - Bank error	Transaction declined due to an error at the bank
Declined	204	Declined - Transaction not supported	Transaction declined as not supported by bank
Declined	205	Declined - Card security code verification failed	Transaction declined as CSC code invalid or not correct for card
Declined	206	Declined - Address verification failed	Transaction declined as Address verification failed for the transaction
Declined	207	Declined - Address and card security code verification failed	Transaction declined as Address and CSC verification failed for the transaction
Declined	270	Declined - 3DSecure authentication failed	Transaction declined as 3DSecure authentication failed and merchant rules do not allow processing to continue
Declined	271	Declined - Card not enrolled in 3DSecure	Transaction declined as card not enrolled in 3DS and merchant rules do not allow processing to continue
Declined	272	Declined - Card enrolled but holder not registered with 3DSecure	Transaction declined as card holder not registered with 3DS and merchant rules do not allow processing to continue
Failed	300	Failed - Error communicating with provider	Transaction failed as could not process via provider
Failed	301	Failed - Error communicating with bank	Transaction failed as provider could not process via bank
Failed	302	Failed - No reply from bank	Transaction failed as there was no response from the bank
Failed	303	Failed - Error with provider	Transaction failed as there was an error with the provider
Failed	304	Failed - Shopping transaction locked (Please try again later)	Transaction failed as the transaction was locked
Failed	305	Failed - Retry limit reached	Transaction failed as the retry limit has been reached

Failed	306	Failed - Duplicate session ID found	Transaction failed as a duplicate session ID has been passed to the gateway
Failed	307	Failed - Card Issuer Institution Returned a Referral Response	Transaction failed as card issuer referred transaction
Failed	353	Failed - Unable to process 3DSecure transaction	Transaction failed as unable to process 3DSecure transaction
	398	Failed - Internal error	Transaction failed due to an internal error
Failed			
Failed	399	Failed - Unknown error	Transaction failed and reason is unknown
Blocked	400	Blocked - Minimum amount violation	Transaction blocked due to minimum transaction amount not met
Blocked	401	Blocked - Maximum amount exceeded	Transaction blocked due to maximum transaction amount being exceeded
Blocked	402	Blocked - Daily amount exceeded	Transaction blocked due to daily transaction amount being exceeded
Blocked	403	Blocked - Daily successful volume exceeded	Transaction blocked due to daily transaction volume being exceeded (successful transactions only)
Blocked	404	Blocked - Monthly amount exceeded	Transaction blocked due to monthly transaction amount being exceeded
Blocked	405	Blocked - Monthly successful volume exceeded	Transaction blocked due to monthly transaction volume being exceeded (successful transactions only)
Blocked	406	Blocked - Card amount exceeded for set period	Transaction blocked due to card amount limit being exceeded for the period set
Blocked	407	Blocked - Card successful volume exceeded for set period	Transaction blocked due to card successful volume being exceeded for the period set
Blocked	408	Blocked - Card unsuccessful volume exceeded for set period	Transaction blocked due to card unsuccessful volume being exceeded for the period set
Blocked	409	Blocked - Card issuing country restricted. Country: {0}	Transaction blocked due to overseas issued card not allowed for merchant
Blocked	410	Blocked - Card is blacklisted	Transaction blocked due to card being on merchant's blacklist
Blocked	470	Blocked - Daily refund amount exceeded	Transaction blocked Refund Daily Transaction Amount
Blocked	498	Blocked - No Rules Set	Transaction blocked No Rules Set
Unknown*	900	Result unknown	Unable to obtain the transaction result
Unknown*	903	3DSecure authentication may not have completed	Did not received response from 3DSecure yet

Appendix B – List of error codes and messages

Error Code	Description
V001	Please specify the payment method
V002	Payment method needs to be standard
V003	Name on card is a required field
V004	Name on card field can only accept 128 characters
V005	Card Name is invalid
V006	Card expiry year is required
V007	Card expiry year should be two digits
V008	Card expiry date needs to be in the future
V009	Card expiry month is required
V0010	Card expiry month should be two digits
V0011	Card expiry month is invalid
V0012	Card expiry date is invalid
V0013	Card CSC is required
V0014	CSC code needs to be 3 or 4 characters
V0015	Card CSC is not valid
V0016	Card number cannot be empty
V0017	Card number is incorrectly formatted
V0018	Payment request Not Found
V0019	Payment request has expired
V0020	Web payments are not enabled for this account
V0021	Transaction failed to process (invalid data), please retry
V0022	Transaction failed to process, please check before retrying
V0023	Card type is not accepted by merchant
V0024	Payment request has already been used