



# Web Integration Direct Post Recurring Card

Integration Guide version 1.0  
19/04/2016

## Table of Contents

Overview.....	1
<b>Introduction</b> .....	1
<b>Checking that Recurring card payments is enabled for your account</b> .....	1
<b>Direct Post Recurring Card Payments Workflow</b> .....	2
<b>Overview</b> .....	2
<b>Registering the Recurring Card Payment Plan Request</b> .....	3
<b>Authentication</b> .....	3
<b>Register Payment Request Data</b> .....	3
<b>Bad Request Response</b> .....	9
<b>Forbidden Response</b> .....	9
<b>Successful Request Response</b> .....	10
<b>Creating the plan</b> .....	11
<b>POST Data</b> .....	11
<b>Retrieving the Result</b> .....	12
<b>Appendices</b> .....	17
<b>Appendix A - List of error codes and messages</b> .....	17

## Overview

### Introduction

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

Before you use direct post recurring card payments, please check that your account is subscribed to the recurring card payments channel.

### Checking that Recurring card payments is enabled for your account

Before using the direct post recurring card payments for making payments you first need to ensure that Recurring Card 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 Recurring Card 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 Recurring Card Payments Workflow**

### **Overview**

Making a recurring card payment using direct post involves three steps:

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

The following steps provide some context using an example where a customer decides to set a recurring card payment plan from a merchant’s web site.

1. The customer visits the merchant’s website and clicks to set a recurring card plan
2. The merchant’s website registers a recurring card plan request through an API call to Flo2Cash
3. A unique URL is returned to the merchant’s website
4. A card form is rendered to the customer allowing them to enter card data. The form action is set to post directly to the URL returned in step 3
5. The Customer enters their card details before submitting the form
6. The form data including card details are posted directly to the unique URL provided in step 3
7. If the data is valid a recurring card plan 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 card plan
10. The merchant requests the plan result and displays this to the customer

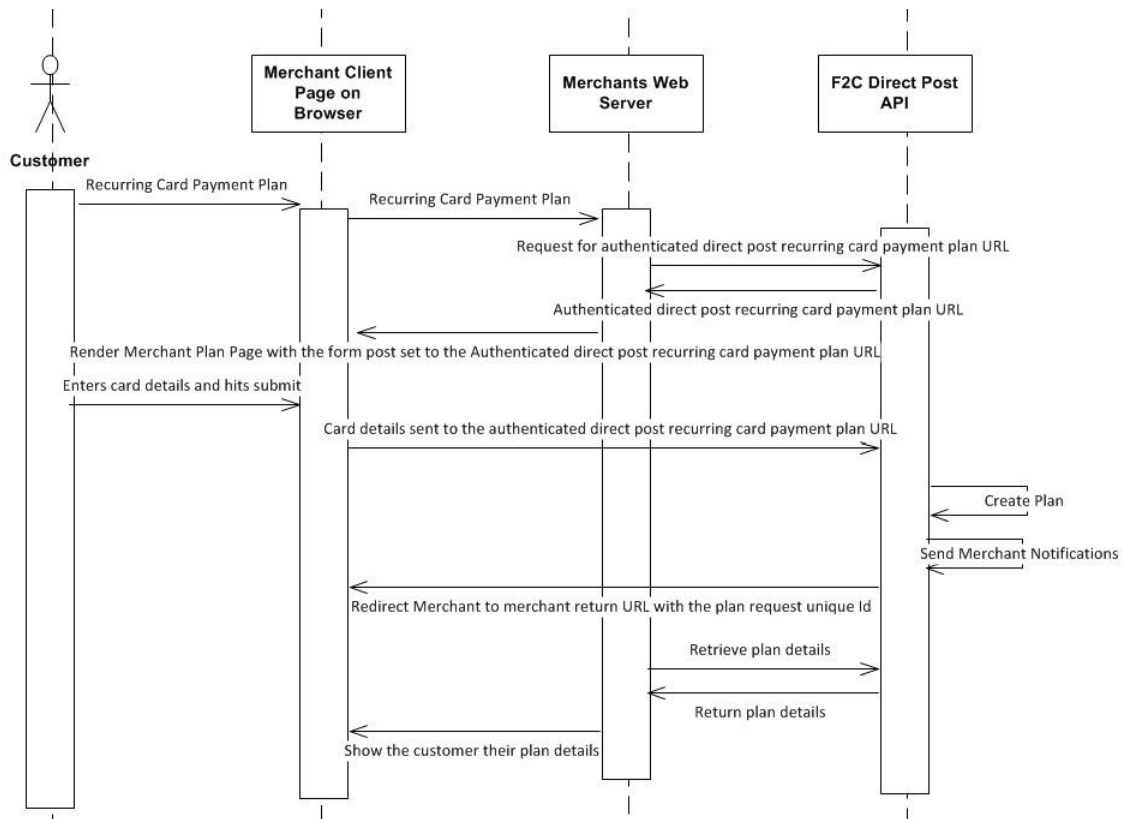


Figure 1 - Direct Post Recurring Card Payment Plan Overview

## Registering the Recurring Card Payment Plan Request

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

<https://secure.flo2cash.co.nz/integration/recurringpayment/card/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 QWxhZGRpbjpvGVuIHNLc2FtZQ==**

Please refer here [http://en.wikipedia.org/wiki/Basic\\_access\\_authentication](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 recurring card payment plan 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
<b>recurringCardPayment</b>		Recurring Card Payment Element	R			
	<b>subAccount</b>	Sub account through which the payment will be processed	R	String	5	Must exist under merchants account
	<b>requestDuration</b>	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.
	<b>returnUrl</b>	The URL the customer will be returned to	R	String	11 - 1024	Valid URL beginning http:// or https://
	<b>notificationUrl</b>	The URL MNS notifications will be sent to	O	String	11 - 1024	Valid URL beginning http:// or https://
	<b>recurringPaymentPlanType</b>	3 – Recurring (on-going recurring payments with fixed amount and fixed frequency) 4 – Instalment (A payment plan with a total amount, which once paid off the plan will automatically finish)	R	Integer	3 – 4	

	<b>frequency</b>	Frequency of the plan 2 – Weekly 3 – Fortnightly 7 – Monthly Fixed Date 10 – Two monthly Fixed Date 11 – Three monthly Fixed Date 12 – Six monthly Fixed Date 13 – Twelve monthly Fixed Date	R	Integer	2-13	
	<b>initialPaymentDate</b>	Payment date for a	O	Date	YYYY-	Need to be at least
	<b>initialPaymentDate</b>	initial payment before the plan starts			MM-DD	tomorrow
	<b>initialPaymentAmount</b>	Amount for the initial payment before the plan starts	O	Decimal	0.01 – n	Must be greater than 0
	<b>planStartDate</b>	Start date of the plan	R	Date	YYYY-MM-DD	Needs to be at least tomorrow
	<b>perPaymentAmount</b>	Amount charged every time a payment is executed from the plan	R	Decimal	0.01 – n	Must be greater than 0
	<b>totalInstalmentAmount</b>	Amount for the instalment	O/R	Decimal	0.01 – n	Must be greater than initial payment amount and per plan amount. <b>Required if the plan type is instalment.</b>
	<b>currency</b>	Currency of payment	R	String	3	ISO currency code - <b>NZD</b> is the only valid value at present
	<b>reference</b>	Merchant defined reference for the payment	O	String	1 – 50	
	<b>particulars</b>	Merchant defined particulars for the payment	O	String	1 – 50	

<b>recurringCardPayment.customer</b>		Customer Element	O			
	<b>companyName</b>	Name of the company	O	String	1 - 50	
	<b>title</b>	Salutation title	O	String	1 - 50	
	<b>firstName</b>	First name	O	String	1 - 128	
	<b>lastName</b>	Last name	O		1 - 128	
	<b>dateOfBirth</b>	Date of birth	O	String	8	Formatted YYYYMMDD
	<b>homePhone</b>	Home phone	O	String	1 - 50	
	<b>workPhone</b>	Work phone	O	String	1 - 50	
	<b>mobilePhone</b>	Mobile phone	O	String	1 - 50	
	<b>fax</b>	Fax number	O	String	1 - 50	
	<b>email</b>	Email address	O	String	1 - 256	
	<b>addressLine1</b>	First line of address	O	String	1 - 128	
	<b>addressLine2</b>	Second line of address	O	String	1 - 128	
	<b>addressLine3</b>	Third line of address	O	String	1 - 128	
	<b>addressSuburb</b>	Suburb of address	O	String	1 - 50	
	<b>addressCity</b>	City of address	O	String	1 - 50	
	<b>addressCountry</b>	Country of address	R	String	2	ISO Alpha 2 Country Code <a href="http://en.wikipedia.org/wiki/ISO_31661_alpha-2">http://en.wikipedia.org/wiki/ISO_31661_alpha-2</a>
	<b>addressPostcode</b>	Postcode of address	O	String	1 - 50	
<b>recurringCardPayment.customData</b>		CustomData Element	O			
<b>recurringCardPayment.customData.data</b>		Data Element	R		1 - 20	
	<b>name</b>	Merchant defined name	R		1 - 50	
	<b>value</b>	Merchant defined value	R		0 - 255	

### XML Example

```

<recurringCardPayment>
  <subAccount>70040</subAccount>
  <requestDuration>100</requestDuration>
  <notificationUrl>http://www.example.org/notification</notificationUrl>
  <returnUrl>http://www.example.com/return</returnUrl>
  <recurringPaymentPlanType>4</recurringPaymentPlanType>
  <frequency>3</frequency>
  <initialPaymentDate>20151001</initialPaymentDate>
  <initialPaymentAmount>1</initialPaymentAmount>
  <planStartDate>20151101</planStartDate>
  <perPaymentAmount>10</perPaymentAmount>
  <totalInstalmentAmount>2000</totalInstalmentAmount >
  <currency>NZD</currency>
  <reference>Ref</reference>
  <particulars>Par</particulars>
  <customer>
    <companyName>Company</companyName>
    <title>Mr</title>
    <firstName>First</firstName>
    <lastName>Last</lastName>    <dateOfBirth>19880412</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>

  <customData>
    <data>
      <name>N1</name>
      <value>V1</value>
    </data>
    <data>
      <name>N2</name>
      <value>V2</value>
    </data>
  </customData>
</recurringCardPayment>

```

## JSON Example

```
{
  "subAccount": "70040",
  "requestDuration": "100",
  "notificationUrl": "", http://www.example.org/notification,
  "returnUrl": "", http://www.example.com/return,
  "recurringPaymentPlanType": "4",
  "frequency": "3",
  "initialPaymentDate": "20151001",
  "initialPaymentAmount": "1",
  "planStartDate": "20151101",
  "perPaymentAmount": "10",  "totalInstalmentAmount":
"2000",
  "currency": "NZD",
  "reference": "Ref",
  "particulars": "Par",
  "customer": {
    "companyName": "Company",
    "title": "Mr",
    "firstName": "First",
    "lastName": "Last",
    "dateOfBirth": "19880412",
    "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": "0614"
  },
  "customData": [
    {
      "name": "N1",
      "value": "V1"
    },
    {
      "name": "N2",
      "value": "V2"
    }
  ]
}
```



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
<b>error</b>		Error Element		
	<b>type</b>	Type of error	String	
	<b>httpStatusCode</b>		String	
<b>error.messages</b>		Messages Element		
	<b>message</b>	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>
  <httpStatusCode>400</httpStatusCode>
  <messages>
    <message>Currency is required</message>
  </messages>
</error>
```

#### JSON Example

```
{
  error:{
    type:'BAD REQUEST',
    httpStatusCode:400,
    messages:{
      message:'Currency is required'
    }
  }
}
```

### Forbidden Response

If the merchant is not subscribed to the recurring card payments 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>Recurring card payment is not enabled for this
account</message>
  </messages>
</error>
```

### JSON Example

```
{
  error:{
    type:'FORBIDDEN',
    statusCode:403,
    messages:{
      message:'Recurring card payment is 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

```
<recurringCardPlanRequest>
  <id>57F31EFE-01AE-4D80-8EB5-7A16D5979DB5</id>

  <url>https://secure.flo2cash.co.nz/integration/recurringpayment/card/dp/57F
31EFE-01AE-4D80-8EB5-7A16D5979DB5</url>
  <expiryDate>2015-01-22T15:37:00</expiryDate>
</ recurringCardPlanRequest >
```

### JSON Example

```
{
  recurringCardPlanRequest:{
    id:'57F31EFE-01AE-4D80-8EB5-7A16D5979DB5',
    url:'https://secure.flo2cash.co.nz/integration/payments/dp/57F31EFE-01AE-
4D80-8EB5-7A16D5979DB5'
    expiryDate:'2015-01-22T15:37:00',
  }
}
```

## Creating the plan

Once a request has been registered and a URL returned, the card details 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 card 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

HTML Form Field Name	Description	Required / Optional	Type	Min-Max	Comments
<b>CardNumber</b>	Full number (PAN) of card to charge	R	String	20	<ul style="list-style-type: none"> <li>• Digits only</li> <li>• Passes Luhn check</li> <li>• Accepted card scheme</li> </ul>
<b>CardCSC</b>	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
<b>CardExpiryMonth</b>	Month the card expires in	R	String	2	Should be a valid month
<b>CardExpiryYear</b>	Year the card expires in	R	String	2	Valid year
<b>NameOnCard</b>	Name of the card holder as it appears on the card	R	String	1- 128	

Once the plan is processed, the user will be redirected to the return URL with the plan request 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>

## HTML Example

```
<html>
  <head></head>

  <body>

    <form
action="https://secure.flo2cash.co.nz/integration/recurringpayments/card/dp
/b2097104-d57c-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>
```

## Retrieving the Result

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

- Validation failed
- Processing error - can be retried
- Processing error - check Merchant Console
- Plan Created

The user needs to make a HTTP GET request to

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

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

**Card Plan Resource**

The resource fields that get returned are explained below.

Element	Child Element	Description	Type
<b>cardPlan</b>		payment Element	
	<b>id</b>	Unique Identifier to process the payment	Guid
	<b>url</b>	Unique Url to set the merchant’s Form Post to	String
	<b>expiryDate</b>	Date/Time of when the unique payment Url expires	DateTime
	<b>requestDuration</b>	The number of minutes the payment request is active before expiring	Integer
	<b>merchant</b>	Merchant Id for the request	String
	<b>subAccount</b>	Sub account through which the payment will be	String

		processed	
	<b>returnUrl</b>	The URL the customer will be returned to	String
	<b>notificationUrl</b>	The URL MNS notifications will be sent to	String
	<b>recurringPaymentPlanType</b>	Plan type for the recurring plan	Integer
	<b>frequency</b>	Frequency of the plan	Integer
	<b>initialPaymentDate</b>	Initial payment date	Date YYYYMM DD
	<b>initialPaymentAmount</b>	Initial payment amount	Decimal
	<b>planStartDate</b>	Start date of the plan	Date YYYYMM DD
	<b>perPaymentAmount</b>	Amount per payment for the plan	Decimal
	<b>totalInstalmentAmount</b>	Instalment amount if the plan type is instalment	Decimal
	<b>currency</b>	Currency of payment	String
	<b>reference</b>	Merchant defined reference for the payment	String
	<b>particulars</b>	Merchant defined particulars for the payment	String
<b>cardPlan.customer</b>		Customer Element	
	<b>companyName</b>	Name of the company	String
	<b>title</b>	Salutation title	String
	<b>firstName</b>	First name	String

	<b>lastName</b>	Last name	
	<b>dateOfBirth</b>	Date of birth	String
	<b>homePhone</b>	Home phone	String
	<b>workPhone</b>	Work phone	String
	<b>mobilePhone</b>	Mobile phone	String
	<b>email</b>	Email Address	String
	<b>fax</b>	Fax Number	String
	<b>addressLine1</b>	First line of address	String
	<b>addressLine2</b>	Second line of address	String
	<b>addressLine3</b>	Third line of address	String
	<b>addressSuburb</b>	Suburb of address	String
	<b>addressCity</b>	City of address	String
	<b>addressCountry</b>	Country of address	String
	<b>addressPostcode</b>	Postcode of address	String
<b>cardPlan.custom Data</b>		CustomData Element	
<b>cardPlan.custom Data.data</b>		Data Element	
	<b>name</b>	Merchant defined name	
	<b>value</b>	Merchant defined value	
<b>cardPlan.processingResult</b>			
	<b>status</b>	Status for the request, please refer to appendix A	Integer
	<b>errorCode</b>	A unique error code indicating the error that has occurred	String
	<b>errorMessage</b>	A text description of the error code	String
	<b>planId</b>	Card plan id	int

### 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 A.

#### 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.

#### Request Processed (5)

If the data is valid then the request will be processed and the result will be available in the following field:

PlanId: Id of the card plan

#### XML Example

```
<cardPlan>
<links>
<id>7cebfe5d-48e3-40ff-9df3-a1fe65cefba7</id>
<url> https://qa.stage.flo2cash.co.nz/integration/recurringpayment/card/dp/7
cebfe5d-48e3-40ff-9df3-a1fe65cefba7</url>
<expiryDate>2015-07-30T12:36:56</expiryDate>
<requestDuration>100</requestDuration>
<merchant>90034</merchant>
<subAccount>70040</subAccount>
<returnUrl>http://www.example.com/return</returnUrl>
<notificationUrl>http://www.example.org/notification</notificationUrl>
<recurringPaymentPlanType>4</recurringPaymentPlanType>
<frequency>3</frequency>
<initialPaymentDate>20151001</initialPaymentDate>
<initialPaymentAmount>1.00</initialPaymentAmount>
<planStartDate>20151101</planStartDate>
<perPaymentAmount>10.00</perPaymentAmount>
<totalInstalmentAmount>2000.00</totalInstalmentAmount>
<currency>NZD</currency>
<reference>Ref</reference> <particulars>Par</particulars>
<customer>
<companyName>Company</companyName>
<title>Mr</title>
<firstName>First</firstName>
<lastName>Last</lastName>
```

```

<dateOfBirth>19880412</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>
<customData>
<data>
<name>N1</name>
<value>V1</value>
</data>
<data>
<name>N2</name>
<value>V2</value>
</data>
</customData>
<processingResult>
<status>5</status>
<errorCode xsi:nil="true" />
<errorMessage xsi:nil="true" />
<planId>827</planId>
</processingResult>
</cardPlan>

```

## JSON EXAMPLE

```

{
  "links": [],
  "id": "7cebfe5d-48e3-40ff-9df3-a1fe65cefba7",
  "url":
  "https://qa.stage.flo2cash.co.nz/integration/recurringpayment/card/dp/7cebfe5d-48e3-40ff-9df3-a1fe65cefba7",
  "expiryDate": "2015-07-30T12:36:56",
  "requestDuration": "100",
  "merchant": "90034",
  "subAccount": "70040",
  "returnUrl": "http://www.example.com/return",
  "notificationUrl": "http://www.example.org/notification",
  "recurringPaymentPlanType": "4",
  "frequency": "3",
  "initialPaymentDate": "20151001",
  "initialPaymentAmount": "1.00",
  "planStartDate": "20151101",
  "perPaymentAmount": "10.00",
  "totalInstalmentAmount": "2000.00",
  "currency": "NZD",

```



```

"reference": "Ref",
"particulars": "Par",
"customer": {
  "companyName": "Company",
  "title": "Mr",
  "firstName": "First",
  "lastName": "Last",
  "dateOfBirth": "19880412",
  "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": "0614"
},
"customData": [
  {
    "name": "N1",
    "value": "V1"
  },
  {
    "name": "N2",
    "value": "V2"
  }
],
"processingResult": {
  "status": "5",
  "planId": "827"
}
}
    
```

## Appendices

### Appendix A - 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

<b>V009</b>	Card expiry month is required
<b>V0010</b>	Card expiry month should be two digits
<b>V0011</b>	Card expiry month is invalid
<b>V0012</b>	Card expiry date is invalid
<b>V0013</b>	Card CSC is required
<b>V0014</b>	CSC code needs to be 3 or 4 characters
<b>V0015</b>	Card CSC is not valid
<b>V0016</b>	Card number cannot be empty
<b>V0017</b>	Card number is incorrectly formatted
<b>V0018</b>	Payment request Not Found
<b>V0019</b>	Payment request has expired
<b>V0020</b>	Web payments are not enabled for this account
<b>V0021</b>	Transaction failed to process (invalid data), please retry
<b>V0022</b>	Transaction failed to process, please check before retrying
<b>V0023</b>	Card type is not accepted by merchant
<b>V0024</b>	Payment request has already been used
<b>V0025</b>	3DS was unsuccessful
<b>V0026</b>	Plan failed to be created (invalid data), please retry
<b>V0027</b>	Plan failed to be created, please check before retrying
<b>V0028</b>	Bank detail is not valid
<b>V0029</b>	Token request has already been used
<b>V0030</b>	Card plan request has already been used
<b>V0031</b>	Direct debit plan request has already been used