BPOINT API

Version 2

May 2015

Contents

1 API Basics

1.1 Overview

The BPOINT Payment Connector offers a way of interfacing with the BPOINT platform programmatically with a REST interface using HTTP over SSL. Requests are submitted to process transactions, create and update entities such as DataVault tokens, and perform searches.

Requests to the API are categorised by the different areas of functionality of the BPOINT platform. The areas currently supported are transaction processing and DataVault token processing. Each area has its own URL endpoint. Different HTTP methods imply different operations, for example retrieval of information from the database is typically handled by the GET operation, while updates are handled by either POST or PUT requests.

For nearly all operations, requests and responses are defined within a JavaScript Object Notation (JSON) object. Where appropriate the Payment Connector also supports WebForm (HTTP POST) requests for some operations allowing seamless integration with web applications.

Errors from the API are handled by both HTTP Status Codes and API level response codes. HTTP Status Code errors handle scenarios such as using incorrect HTTP verbs or incorrect URLs. API response codes handle validation, system errors and invalid authentication details.

1.2 Getting Started

1.2.1 API Access

To access the API, you must:

- have the BPOINT Checkout or BPOINT Enterprise package
- create an API user via Back Office > ADMIN > User Management

1.2.2 Base API URLs

Production:	**https://www.bpoint.com.au/webapi/v2**
UAT:	**https://bpoint-uat.premier.com.au/webapi/v2**

The URL your application will access depends on the functionality required. For example, all functionality related to transaction processing uses the following endpoints:

Production:	https://www.bpoint.com.au/webapi/v2/txns
UAT:	https://bpoint-uat.premier.com.au/webapi/v2/txns

1.3 Integration methods

The API supports two integration methods: Direct and Browser. In cases where the web application needs to perform operations that handle card data (such as transaction processing) and the merchant does not want to the card data to be posted to their own web servers, the application can be integrated with the API using the **Browser** integration method.

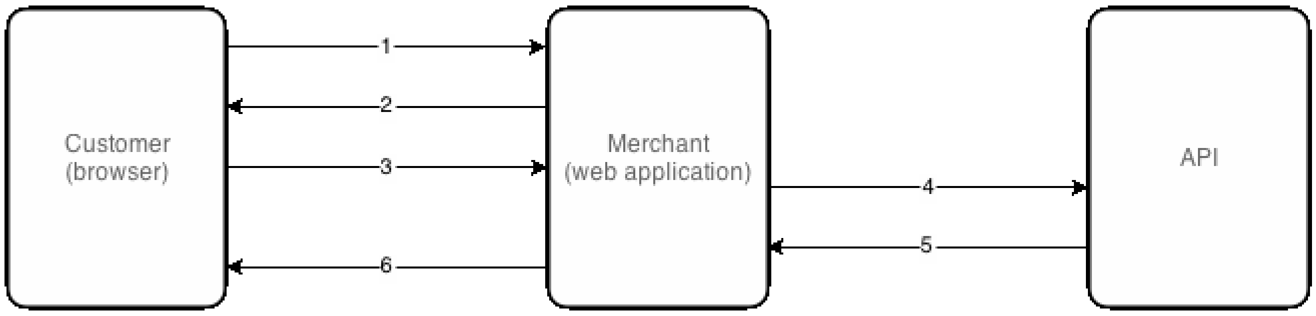
1.3.1 Direct Integration

In this integration method, the merchant's application interfaces directly with the API. The API supports this method for ALL operations.

Every request is sent directly from the merchant's application to the API and every request must contain authentication details. Since the communication is between the merchant's web application and the API, the credentials are not exposed to any 3rd parties.

The response is sent directly back to the merchant's application.

The following diagram shows a typical "process transaction" flow using the **Direct** integration method.



1. Card holder requests the merchant's payment page.
2. Merchant's web application renders payment page to the card holder.
3. Card holder then enters their payment details, including card data, and clicks on the Submit button. The payment details and card data are then sent to the merchant's web application.
4. Merchant's web application sends a "process transaction" request to the API.
5. The API processes the payment in real time and returns the response to the merchant's web application. The response contains the transaction result.
6. Merchant's web application then renders a receipt page to the card holder.

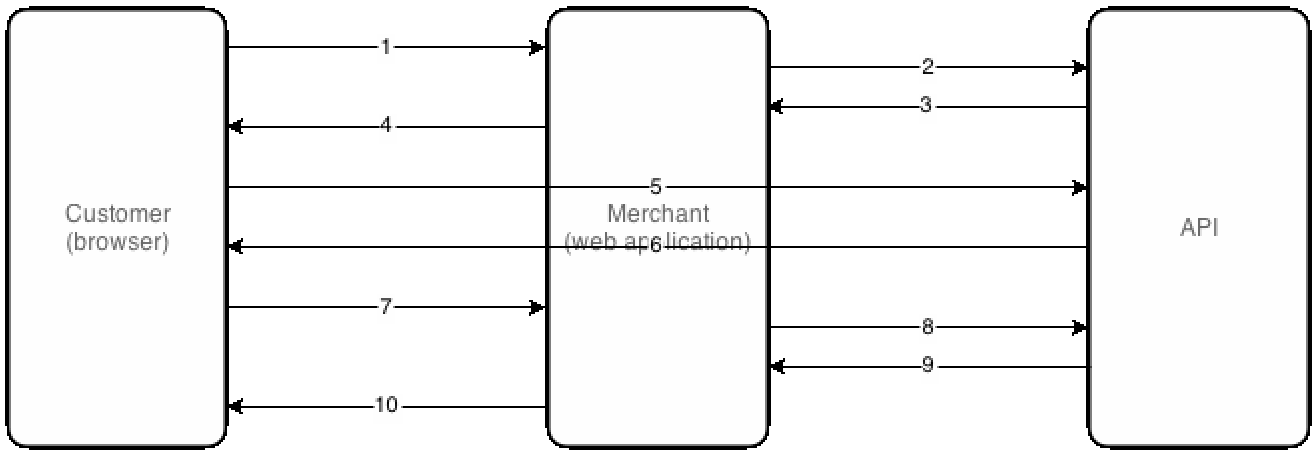
IMPORTANT:
Using the **Direct** integration method for operations where card data is handled means that card data will go through the merchant's system.

1.3.2 Browser Integration

In this integration method the browser posts the data directly to the API. A redirect mechanism can be used or the inbuilt JavaScript library provides a seamless integration. The API supports the **Browser** integration method for any operation that accepts card data.

Since the card data is submitted directly from the card holder's internet browser to the API over SSL the merchant system does not need to handle the card data.

The following diagram shows a typical "process transaction" flow using the **Browser** integration method for operations that handle card data.



1. Card holder requests the merchant's payment page.
2. Merchant's web application sends a "create AuthKey" request to the API (using **Direct** integration method).
3. API creates a one-time AuthKey and returns it to the merchant's web application.
4. Merchant's web application renders the payment page to the card holder. The AuthKey is included as a hidden field on the payment page.
5. Card holder enters their payment details, including card data, and clicks on the Submit button. The payment details and card data are sent directly to the API, bypassing the merchant's web application (using **Browser** integration method).
6. API processes a payment in real time and returns a "redirection" response directly to the customer's browser. The redirection is to the merchant's receipt page URL. The API response and ResultKey are part of the redirection URL. The result of the payment is not included.
7. Browser redirects to the merchant's receipt page.
8. Merchant's web application retrieves the API response and ResultKey from the query string and sends a "transaction result lookup" request to the API (using **Direct** integration method).
9. API retrieves the transaction result and returns it back to the merchant's web application.
10. Merchant's web application then renders the receipt page to the card holder.

There are 2 ways to implement the **Browser** integration method into a web application.

1. WebForms – merchant's web page uses html form data POST request to submit details to the API. The AuthKey should be included as a hidden field on the form and submitted with the request.
2. BPOINT JavaScript library – a web page can include the BPOINT JavaScript file and let it handle request and response processing.

2 Requests and Responses

2.1 Authentication

2.1.1 Direct Integration

In Direct Integration Method requests use an ***"Authorization"*** header to pass a base64 encoded string containing the API user credentials. The string is formatted as follows:

username|merchantnumber:password

For example, the HTTP header for above credentials string is:

Authorization: dXNlcm5hbWV8bWVyY2hhbnRudW1iZXI6cGFzc3dvcmQ=

2.1.2 Browser Integration

Since these requests are initiated from the customer's internet browser, the API user credentials cannot be used. Instead these requests utilise a one time session ID called an **AuthKey**, The merchant's web application needs to request an AuthKey from the API prior to rendering a web page which interfaces with the API.

2.2 HTTP Status Codes

There are only 2 HTTP status codes that indicate success:

- **200** – Request completed successfully. HTTP response body will contain an API JSON response object, used to determine the result of the operation.
- **302** – Redirection response (returned only for requests made using the WebForms **Browser** integration method), indicates that the request was completed successfully. The browser will redirect to the URL included in the HTTP response. The API response fields will be included in the query string appended to the URL.

Any other HTTP status code returned indicates an unsuccessful request.

2.3 API Responses

All API responses share the following common fields:

Name:	Comments:	
ResponseCode	**Description:**	Numeric code returned from the API to indicate whether request was successful (0) or not (any non 0 value). API responses are listed in section 6.1
	Format:	Integer
	Example:	1
ResponseText	**Description:**	Text returned from the API to provide the description of the response. API responses are listed in section 6.1
	Format:	String
	Constraints:	ANS[all], MAXLEN=500
	Example:	Invalid login details

2.3.1 Direct Integration

In the Direct Integration Method, all JSON API responses contain the ResponseCode and the ResponseText within the APIResponse object. Assuming that the ResponseCode indicates a successful response, the response object will also include the detailed object representing the result of the operation, eg: transaction response.

Example:

```
{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        },
        "TxnResp" : {
                ...
        }
}
```

2.3.2 Browser Integration

In the Browser integration method, all requests and responses are handled by the customer's web browser. Due to this, detailed response objects are never included in the result of the operation. Instead, the ResultKey field is returned to allow the merchant to look up the detailed response using the Direct integration method, and determine what information to display to their customer on the merchant receipt page:

Name:	**Comments:**	
ResultKey	**Description:**	Unique key that the merchant must use to look up the detailed result of the operation. The ResultKey is returned only if the API response code indicates success.
	Format:	String
	Constraints:	ANS[all], MAXLEN=500
	Example:	5df3c8cd-f927-4dc0-8ea8-c9a255bee04d

2.3.2.1 WebForms Request

For WebForms requests a response is a HTTP 302 Redirect response to the merchant's receipt page URL. ResponseCode and ResponseText will always be appended to the redirection URL in a query string. Assuming that the ResponseCode indicates a successful response, the query string will also include the ResultKey.

Redirection URL query string example:
?ResponseCode=0&ResponseText=Success&ResultKey=5df3c8cd-f927-4dc0-8ea8-c9a255bee04d

2.4 Common JSON Objects

2.4.1 Data Types and Constraints

Data type:	**Constraint:**	**Description:**
String	**A**	Alphabetic characters
	N	Numeric characters
	S []	Special characters (followed by a list of allowed characters inside [])
	MINLEN	Minimum length

	MAXLEN	Maximum length
Integer	**MIN**	Minimum value
	MAX	Maximum value

2.4.2 Common Request Objects

2.4.2.1 Card Details

Name:	**Comments:**	
CardDetails	**Description:**	JSON object containing card details
CardHolderName	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Card holder name String OPTIONAL ANS [all], MAXLEN=50 Yes John Smith
CardNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card number String MANDATORY N, MINLEN=13, MAXLEN=16 Yes 5123456789012346
Cvn	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Card verification number String MANDATORY N, MINLEN=3, MAXLEN=4 Yes 123
ExpiryDate	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date In MMYY format String MANDATORY N, MINLEN=4, MAXLEN=4 Yes 0517

2.4.2.2 Card Details (Browser Integration)

Name:	**Comments:**	
CardDetails	**Description:**	JSON object containing card details. Used only in Browser integration, allows card expiry date to be submitted as two separate fields.
CardHolderName	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Card holder name String OPTIONAL ANS [all], MAXLEN=50 Yes John Smith
CardNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card number String MANDATORY N, MINLEN=13, MAXLEN=16 Yes 5123456789012346
Cvn	**Description:** **Format:** **Constraints:**	Card verification number String MANDATORY N, MINLEN=3, MAXLEN=4

	Validated: **Example:**	Yes 123
ExpiryDateMonth	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date month In MM format String MANDATORY N, MINLEN=2, MAXLEN=2 Yes 05
ExpiryDateYear	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date year In YY format String MANDATORY N, MINLEN=2, MAXLEN=2 Yes 17

2.4.2.3 Bank Account Details

Name:	**Comments:**	
BankAccountDetails	**Description:**	JSON object containing bank account details
AccountName	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Bank account name String MANDATORY ANS [EBCDIC character set], MINLEN=1, MAXLEN=32 Yes John Smith
AccountNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Bank account number String MANDATORY AN, MINLEN=3, MAXLEN=9 Yes 00123
BSBNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	BSB number String MANDATORY N, MINLEN=6, MAXLEN=6 Yes 061200

2.4.2.4 DVToken Request

Name:	**Comments:**	
DVTokenReq	**Description:**	JSON object containing DataVault token request details
BankAccountDetails	See section 2.4.2.3 **Constraints:**	 MANDATORY if CardDetails are not present
CardDetails	See section 2.4.2.1 **Constraints:**	 MANDATORY if BankAccountDetails are not present. CVN field is NOT REQUIRED and will be ignored if present.
Crn1	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 1 String MANDATORY ANS [], MAXLEN=50 Yes Customer 1234
Crn2	**Description:** **Format:**	Customer reference number 2 String

	Constraints: **Validated:** **Example:**	OPTIONAL ANS [], MAXLEN=50 Yes Invoice 987
Crn3	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 3 String OPTIONAL ANS [], MAXLEN=50 Yes customer@email.com
EmailAddress	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer's email address String OPTIONAL ANS [], MAXLEN=250 Yes customer@email.com

2.4.3 Common Response Objects

2.4.3.1 Card Details

Name:	**Comments:**	
CardDetails	**Description:**	JSON object containing the masked credit card details, used only in API responses.
CardHolderName	**Description:** **Format:** **Example:**	Card holder name String John Smith
ExpiryDate	**Description:** **Format:** **Example:**	Credit card expiry date In MMYY format String 0517
MaskedCardNumber	**Description:** **Format:** **Example:**	Masked credit card number. First 6 digits of card number, followed by "…" and then by last 3 digits of card number. String 444433…111

2.4.3.2 Bank Account Details

Name:	**Comments:**	
BankAccountDetails	**Description:**	JSON object containing masked bank account details, used only in API responses.
AccountName	**Description:** **Format:** **Example:**	Bank account name String John Smith
AccountNumber	**Description:** **Format:** **Example:**	Bank account number String 00123
BSBNumber	**Description:** **Format:** **Example:**	BSB number String 061200
TruncatedAccountNumber	**Description:** **Format:** **Example:**	Truncated bank account details. BSB number, followed by "###" and then by last 3 digits of bank account number. String 061200###123

2.4.3.3 CVN Result

Name:	Comments:	
CVNResult	**Description:**	JSON object containing CVN verification result.
CVNResultCode	**Description:** **Format:** **Example:**	Returned by the Card Issuer to show the level of match that occurred with the CVN check. See section 6.7 String M

2.4.3.4 ThreeDS Response

Name:	Comments:	
ThreeDSResponse	**Description:**	JSON object containing Three DS verification result.
Eci	**Description:** **Format:** **Example:**	Electronic Commerce Indicator provides the Three DS authentication result String 05
Enrolled	**Description:** **Format:** **Example:**	Indicates if card is enrolled for Three DS String Y
Status	**Description:** **Format:** **Example:**	Three DS Status which is only returned if authentication was attempted. String Y
VerifySecurityLevel	**Description:** **Format:** **Example:**	Verification security level generated by the Card Issuer to prove that the cardholder was enrolled and authenticated. This is not generated when the authentication status is "Failure" String 2
VerifyStatus	**Description:** **Format:** **Example:**	Verification status shows whether the payment authentication was successful String
VerifyToken	**Description:** **Format:** **Example:**	Verification token generated by the Card Issuer to prove that the cardholder was authenticated String
VerifyType	**Description:** **Format:** **Example:**	Verification type String 3DS
XId	**Description:** **Format:** **Example:**	XID is a unique identifier for Three DS transactions String

2.4.3.5 Transaction Response

Name:	Comments:	
TxnResp	**Description:**	JSON object containing transaction result.
Action	**Description:** **Format:** **Example:**	Defines what financial transaction was processed String payment
Amount	**Description:** **Format:** **Example:**	Total transaction amount in the lowest denomination for the currency 64 bit integer If AUD, 12000 for $120.00. If JPY, 120 for ¥120
AmountSurcharge	**Description:**	Surcharge amount in cents

	Format: **Example:**	64 bit integer If AUD, 12000 for $120.00. If JPY, 120 for ¥120
AuthoriseId	**Description:** **Format:** **Example:**	Authorise ID issued by the Acquirer for approved transactions. Not all Acquirers return the AuthoriseId even if the transaction is approved. String R01264
BankAccountDetails	See section 2.4.3.2	
BankResponseCode	**Description:** **Format:** **Example:**	Response code returned from the issuing bank String 00
BillerCode	**Description:** **Format:** **Example:**	Biller code String 21
CVNResult	See section 2.4.3.3	
CardDetails	See section 2.4.3.1	
CardType	**Description:** **Format:** **Example:**	Type of card used to process transaction. See section 6.5 String MC
Crn1	**Description:** **Format:** **Example:**	Customer reference number 1 String Customer 1234
Crn2	**Description:** **Format:** **Example:**	Customer reference number 2 String Invoice 987
Crn3	**Description:** **Format:** **Example:**	Customer reference number 3 String customer@email.com
Currency	**Description:**	Reserved for future use
IsCVNPresent	**Description:** **Format:** **Example:**	Indicates whether CVN was present Boolean true
IsThreeDS	**Description:** **Format:** **Example:**	Indicates whether transaction was subject to Three DS verification Boolean true
MerchantNumber	**Description:** **Format:** **Example:**	Merchant number, issued by bank String 5353000000000001
MerchantReference	**Description:** **Format:** **Example:**	Merchant reference number String 98787682
OriginalTxnNumber	**Description:** **Format:** **Example:**	Returned for refunds, reversals and captures. Transaction number of the original transaction. String 123456
ProcessedDateTime	**Description:** **Format:** **Example:**	Date and time when transaction was processed. Returned in ISO8601 format. String 2014-11-05T12:53:27.4738695+11:00
RRN	**Description:** **Format:** **Example:**	Retrieval reference number String 426012419890
ReceiptNumber	**Description:** **Format:** **Example:**	Receipt number String 47006772463
ResponseCode	**Description:** **Format:**	Summary response code. 0 means **Approved** transaction. Any other value indicates Decline or Error. String

	Example:	0
ResponseText	**Description:** **Format:** **Example:**	Description of ResponseCode String Approved
SettlementDate	**Description:** **Format:** **Example:**	Date of settlement. In YYYYMMDD format. String 20140901
Source	**Description:** **Format:** **Example:**	Transaction origin String api
StoreCard	**Description:** **Format:** **Example:**	Flag to indicate whether the cardholder agreed to save their card details. Boolean true
SubType	**Description:** **Format:** **Example:**	Transaction sub type See section 6.4 String single
ThreeDSResponse	See section 2.4.3.4	
TxnNumber	**Description:** **Format:** **Example:**	Unique transaction number. String 1254859
Type	**Description:** **Format:** **Example:**	Transaction type See section 6.3 String ivr

2.4.3.6 DVToken Response

Name:	**Comments:**	
DVTokenResp	**Description:**	JSON object containing DataVault token result.
BankAccountDetails	See section 2.4.3.2	
CardDetails	See section 2.4.3.1	
CardType	**Description:** **Format:** **Example:**	Type of card saved. See section 6.5 String VC
Crn1	**Description:** **Format:** **Example:**	Customer reference number 1 String Customer 1234
Crn2	**Description:** **Format:** **Example:**	Customer reference number 2 String Invoice 987
Crn3	**Description:** **Format:** **Example:**	Customer reference number 3 String customer@email.com
EmailAddress	**Description:** **Format:** **Example:**	Customer’s email address String customer@email.com
DVToken	**Description:** **Format:** **Example:**	System generated DataVault token. DataVault tokens can be used in place of a card number to process transactions using the payment details saved for that DataVault token. String 5999991183131863

3 Transactions

3.1 Process Transaction

Processes a transaction.

POST /txns/

Integration:

- Direct

URL Parameters:

- None

Request Headers:

- Authorisation: base64 encoded API user credentials
- Content-Type: application/json; charset=utf-8

Request JSON Object:

Name:	**Comments:**	
TxnReq	**Description:**	JSON object containing transaction request details.
Action	**Description:**	Defines what financial transaction to process. See section 6.2 For this operation, one of the following values MUST be used: • payment • refund • preauth • capture • reversal • unmatched_refund
	Format:	String
	Constraints:	MANDATORY A
	Validated:	Yes
	Example:	payment
Amount	**Description:**	Transaction amount in the lowest denomination for the currency
	Format:	64 bit integer
	Constraints:	MANDATORY, with the exception of reversals MINVAL=1
	Validated:	Yes
	Example:	If AUD, 12000 for $120.00. If JPY, 120 for ¥120
BillerCode	**Description:**	Biller code
	Format:	String
	Constraints:	OPTIONAL N, MAXLEN=50
	Validated:	Yes
	Example:	21
CardDetails	See section 2.4.2.1	
	Constraints:	MANDATORY when action is payment, preauth or unmatched_refund, otherwise NOT REQUIRED and will be ignored
Crn1	**Description:**	Customer reference number 1
	Format:	String
	Constraints:	MANDATORY ANS [], MAXLEN=50

	Validated: **Example:**	Yes Customer 1234
Crn2	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 2 String OPTIONAL ANS [], MAXLEN=50 Yes Invoice 987
Crn3	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 3 String OPTIONAL ANS [], MAXLEN=50 Yes customer@email.com
Currency	**Description:**	Reserved for future use. Set to null.
Customer	**Description:**	Reserved for future use. Set to null.
MerchantReference	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Merchant reference number String OPTIONAL ANS [], MAXLEN=50 Yes 987654
Order	**Description:**	Reserved for future use. Set to null.
OriginalTxnNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Used for refunds, reversals and captures. For refunds it is the transaction number of the payment or capture being refunded. For reversals it is the transaction number of the payment, refund, preauth, capture or unmatched_refund that is being reversed. For captures it is the transaction number of preauth that is being captured. String MANDATORY when action is refund, reversal or capture, otherwise NOT REQUIRED and will be ignored N, MAXLEN=50 Yes 123456
StoreCard	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Flag to indicate whether the cardholder agrees to save their card details. This flag allows merchant to create a DataVault token from the card details used to process the transaction. Boolean OPTIONAL A, if present must be either “true” or “false” Yes true
SubType	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Defines a sub type for transaction See section 6.4 String MANDATORY A Yes single
Type	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Defines a type for transaction See section 6.3 String MANDATORY A Yes ivr

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
TxnResp	See section 2.4.3.5

Status Codes:

- 200 OK – Request completed successfully

Request:

```
POST https://www.bpoint.com.au/webapi/v2/txns/ HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
Content-Length: 397
Expect: 100-continue
Connection: Keep-Alive

{
        "TxnReq" : {
                "Action" : "payment",
                "Amount" : 19900,
                "BillerCode" : null,
                "CardDetails" : {
                        "CardHolderName" : "John Smith",
                        "CardNumber" : "5123456789012346",
                        "Cvn" : "123",
                        "ExpiryDate" : "0517"
                },
                "Crn1" : "test crn1",
                "Crn2" : "test crn2",
                "Crn3" : "test crn3",
                "Currency" : "AUD",
                "Customer" : null,
                "MerchantReference" : "test merchant ref",
                "Order" : null,
                "OriginalTxnNumber" : null,
                "StoreCard" : false,
                "SubType" : "single",
                "Type" : "internet"
        }
}
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 901
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
Set-Cookie: ASP.NET_SessionId=xhu3ioocforgzgnlxdpffmbg; path=/; HttpOnly
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 01:37:32 GMT

{
        "APIResponse" : {
```

```
                        "ResponseCode" : 0,
                        "ResponseText" : "Success"
            },
            "TxnResp" : {
                        "Action" : "payment",
                        "Amount" : 19900,
                        "AmountSurcharge" : 0,
                        "AmountSurchargeGST" : 0,
                        "AuthoriseId" : "585366",
                        "BankAccountDetails" : null,
                        "BankResponseCode" : "00",
                        "BillerCode" : null,
                        "CVNResult" : {
                                    "CVNResultCode" : "Unsupported"
                        },
                        "CardDetails" : {
                                    "CardHolderName" : "John Smith",
                                    "ExpiryDate" : "0517",
                                    "MaskedCardNumber" : "512345...346"
                        },
                        "CardType" : "MC",
                        "Crn1" : "test crn1",
                        "Crn2" : "test crn2",
                        "Crn3" : "test crn3",
                        "Currency" : null,
                        "IsCVNPresent" : true,
                        "IsThreeDS" : false,
                        "MerchantNumber" : "0000000000000000",
                        "MerchantReference" : "test merchant ref",
                        "OriginalTxnNumber" : null,
                        "ProcessedDateTime" : "2014-11-06T01:37:32.3400000",
                        "RRN" : "431012585366",
                        "ReceiptNumber" : "48344826582",
                        "ResponseCode" : "0",
                        "ResponseText" : "Approved",
                        "SettlementDate" : "20141106",
                        "Source" : "api",
                        "StoreCard" : false,
                        "SubType" : "single",
                        "ThreeDSResponse" : null,
                        "TxnNumber" : "46476582",
                        "Type" : "internet"
            }
}
```

3.2 Retrieve Transaction Result

Retrieves details of a previously processed transaction.

GET /txns/{txnNumber}

Integration:

- Direct

URL Parameters:

- txnNumber – transaction number of a previously processed transaction

Request Headers:

- Authorisation: base64 encoded API user credentials

Request JSON Object:

- None

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
TxnResp	See section 2.4.3.5

Status Codes:

- 200 OK – Request completed successfully

Request:

```
GET https://www.bpoint.com.au/webapi/v2/txns/46476584 HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Host: www.bpoint.com.au
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 899
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 01:46:44 GMT

{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        },
        "TxnResp" : {
                "Action" : "payment",
                "Amount" : 19900,
                "AmountSurcharge" : 0,
                "AmountSurchargeGST" : 0,
                "AuthoriseId" : "591425",
                "BankAccountDetails" : null,
                "BankResponseCode" : "00",
                "BillerCode" : "",
                "CVNResult" : {
                        "CVNResultCode" : "Unsupported"
                },
                "CardDetails" : {
                        "CardHolderName" : "John Smith",
                        "ExpiryDate" : "0517",
                        "MaskedCardNumber" : "512345...346"
                },
                "CardType" : "MC",
                "Crn1" : "test crn1",
                "Crn2" : "test crn2",
                "Crn3" : "test crn3",
                "Currency" : null,
                "IsCVNPresent" : true,
                "IsThreeDS" : false,
                "MerchantNumber" : "0000000000000000",
                "MerchantReference" : "test merchant ref",
                "OriginalTxnNumber" : null,
                "ProcessedDateTime" : "2014-11-05T15:46:42.7600000",
                "RRN" : "431012591425",
                "ReceiptNumber" : "48345006584",
                "ResponseCode" : "0",
                "ResponseText" : "Approved",
                "SettlementDate" : "20141106",
                "Source" : "api",
                "StoreCard" : false,
                "SubType" : "single",
                "ThreeDSResponse" : null,
                "TxnNumber" : "46476584",
                "Type" : "internet"
        }
}
```

3.3 Search Transactions

Performs a search on previously processed transactions.

POST /txns/search

Integration:

- Direct

URL Parameters:

- None

Request Headers:

- Authorisation: base64 encoded API user credentials
- Content-Type: application/json; charset=utf-8

Request JSON Object:

Name:	Comments:	
SearchInput	**Description:**	JSON object containing search details.
Action	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Defines what financial transaction to search for. See section 6.2 String OPTIONAL A No payment
Amount	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Amount in the lowest denomination for the currency. 64 bit integer Can be set to 0 to search for all amounts. OPTIONAL No If AUD, 12000 for $120.00. If JPY, 120 for ¥120
AuthoriseId	**Description:** **Format:** **Constraints:** **Validated:**	Authorise ID issued by the Acquirer for approved transactions. Not all Acquirers return the AuthoriseId even if the transaction is approved. String OPTIONAL AN, MAXLEN=50 No

	Example:	R01264
BankResponseCode	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Response code returned from the issuing bank String OPTIONAL AN, MAXLEN=2 No 00
BillerCode	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Biller code String OPTIONAL N, MAXLEN=50 No 21
CardType	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Type of card used to process transaction. See section 6.5 String OPTIONAL A, MAXLEN=2 No MC
Crn1	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 1 String OPTIONAL ANS [], MAXLEN=50 No Customer 1234
Crn2	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 2 String OPTIONAL ANS [], MAXLEN=50 No Invoice 987
Crn3	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 3 String OPTIONAL ANS [], MAXLEN=50 No customer@email.com
Currency	**Description:**	Reserved for future use. Set to null.
ExpiryDate	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date In MMYY format String OPTIONAL N, MINLEN=4, MAXLEN=4 No 0517
FromDate	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Start date for a date range search, compared against transaction processed date. In ISO8601 format. String OPTIONAL ANS [], MAXLEN=50 No 2014-11-05T12:53:27.4738695+11:00
MaskedCardNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Masked card number. Can also be masked bank account details if searching for bank account transactions, eg: 061200###123 String OPTIONAL ANS [.#], MAXLEN=12 No 512345…346
MerchantReference	**Description:** **Format:**	Merchant reference number String

	Constraints:	OPTIONAL ANS [], MAXLEN=50
	Validated:	No
	Example:	987654
RRN	**Description:**	Retrieval reference number
	Format:	String
	Constraints:	OPTIONAL AN, MAXLEN=50
	Validated:	No
	Example:	426012419890
ReceiptNumber	**Description:**	Receipt number
	Format:	String
	Constraints:	OPTIONAL N, MAXLEN=50
	Validated:	No
	Example:	47006772463
ResponseCode	**Description:**	Summary response code. 0 means **Approved** transaction. Any other value indicates Decline or Error.
	Format:	String
	Constraints:	OPTIONAL ANS [_], MAXLEN=50
	Validated:	No
	Example:	0
SettlementDate	**Description:**	Date of settlement. In YYYYMMDD format.
	Format:	String
	Constraints:	OPTIONAL N, MAXLEN=8
	Validated:	No
	Example:	20140901
Source	**Description:**	Transaction origin. See section 6.6
	Format:	String
	Constraints:	OPTIONAL A, MAXLEN=50
	Validated:	No
	Example:	api
ToDate	**Description:**	End date for a date range search, compared against transaction processed date. In ISO8601 format.
	Format:	String
	Constraints:	OPTIONAL ANS [], MAXLEN=50
	Validated:	No
	Example:	2014-11-05T12:53:27.4738695+11:00
TxnNumber	**Description:**	Unique transaction number
	Format:	String
	Constraints:	OPTIONAL N, MAXLEN=50
	Validated:	No
	Example:	1254859

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
TxnRespList	Array of Transaction Response objects. See section 2.4.3.5

Status Codes:

- 200 OK – Request completed successfully

Request:

```
POST https://www.bpoint.com.au/webapi/v2/txns/search HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
Content-Length: 435
Expect: 100-continue

{
        "SearchInput" : {
                "Action" : null,
                "Amount" : 0,
                "AuthoriseId" : null,
                "BankResponseCode" : null,
                "BillerCode" : null,
                "CardType" : null,
                "Crn1" : null,
                "Crn2" : null,
                "Crn3" : null,
                "Currency" : null,
                "ExpiryDate" : null,
                "FromDate" : "2014-11-05T12:53:27.4738695+11:00",
                "MaskedCardNumber" : null,
                "MerchantReference" : null,
                "RRN" : null,
                "ReceiptNumber" : null,
                "ResponseCode" : null,
                "SettlementDate" : null,
                "Source" : null,
                "ToDate" : "2014-11-06T12:53:27.4738695+11:00",
                "TxnNumber" : null
        }
}
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 12750
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
Set-Cookie: ASP.NET_SessionId=11faxeluhcdgf4od5n5ydspl; path=/; HttpOnly
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 01:53:28 GMT

{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        },
        "TxnRespList" : [{
                        "Action" : "payment",
                        "Amount" : 1000,
                        "AmountSurcharge" : 0,
                        "AmountSurchargeGST" : 0,
                        "AuthoriseId" : "051301040945",
                        "BankAccountDetails" : null,
                        "BankResponseCode" : "00",
                "BillerCode" : null,
                "CVNResult" : null,
                "CardDetails" : {
                        "CardHolderName" : null,
                        "ExpiryDate" : "9900",
                        "MaskedCardNumber" : "512345...346"
                },
                "CardType" : "MC",
                "Crn1" : "123456",
                "Crn2" : "",
                "Crn3" : "",
                "Currency" : null,
                "IsCVNPresent" : false,
                "IsThreeDS" : false,
                "MerchantNumber" : "0000000000000000",
                "MerchantReference" : "test",
                "OriginalTxnNumber" : null,
                "ProcessedDateTime" : "2014-11-05T03:01:03.8230000",
                "RRN" : null,
                "ReceiptNumber" : "48330646552",
                "ResponseCode" : "0",
                "ResponseText" : "Approved",
                "SettlementDate" : "20141106",
                "Source" : "backoffice",
                "StoreCard" : false,
                "SubType" : "recurring",
                "ThreeDSResponse" : null,
                "TxnNumber" : "46476552",
                "Type" : "mailorder"
        }, ..., {
                "Action" : "payment",
                "Amount" : 19900,
                "AmountSurcharge" : 0,
                "AmountSurchargeGST" : 0,
                "AuthoriseId" : "591425",
                "BankAccountDetails" : null,
                "BankResponseCode" : "00",
                "BillerCode" : null,
                "CVNResult" : {
                        "CVNResultCode" : "Unsupported"
                },
                "CardDetails" : {
                        "CardHolderName" : null,
                        "ExpiryDate" : "0517",
                        "MaskedCardNumber" : "512345...346"
                },
                "CardType" : "MC",
                "Crn1" : "test crn1",
                "Crn2" : "test crn2",
                "Crn3" : "test crn3",
                "Currency" : null,
                "IsCVNPresent" : true,
                "IsThreeDS" : false,
                "MerchantNumber" : "0000000000000000",
                "MerchantReference" : "test merchant ref",
                "OriginalTxnNumber" : null,
                "ProcessedDateTime" : "2014-11-05T15:46:42.7600000",
                "RRN" : "431012591425",
                "ReceiptNumber" : "48345006584",
                "ResponseCode" : "0",
                "ResponseText" : "Approved",
                "SettlementDate" : "20141106",
                "Source" : "api",
                "StoreCard" : false,
                "SubType" : "single",
                "ThreeDSResponse" : null,
                                "TxnNumber" : "46476584",
                                "Type" : "internet"
                      }
            ]
}
```

3.4 Create AuthKey to Process Transaction

Creates one-time AuthKey to be used with “Process Transaction” operation when using **Browser** integration method.

This operation serves the following functions:

- Authenticates the merchant by username, merchant number and password. If these details are not correct, the AuthKey will not be generated.
- Creates a unique AuthKey to allow a merchant’s customer to process a payment.
- AuthKey prevents processing of duplicate payments.
- Allows merchant to “lock in” certain transaction details and prevents the details from being modified even if submitted in subsequent “Process transaction” request. These fields include:
    - Amount
    - BillerCode
    - Crn1
    - Crn2
    - Crn3
    - MerchantReference
- AuthKey is valid only for a predefined period of time (20 minutes). If the transaction request is not attempted within that time merchant’s application will need to request a new AuthKey.

POST /txns/processtxnauthkey

Integration:

- Direct

URL Parameters:

- None

Request Headers:

- Authorisation: base64 encoded API user credentials
- Content-Type: application/json; charset=utf-8

Request JSON Object:

Name:	Comments:
ProcessTxnData	
Action	**Description:** Defines what financial transaction to process. Must be one of the following values: • payment • preauth MANDATORY **Format:** String **Constraints:** A **Validated:** Yes **Example:** payment **Sub-elements:** No
Amount	**Description:** Transaction amount in the lowest denomination for the currency.

	Format: **Constraints:** **Validated:** **Example:** **Sub-elements:**	OPTIONAL 64 bit integer MINVAL=1 Yes If AUD, 12000 for $120.00. If JPY, 120 for ¥120 No
BillerCode	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Biller code. String OPTIONAL N, MAXLEN=50 Yes 21
Crn1	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 1 String OPTIONAL ANS [], MAXLEN=50 Yes Customer 1234
Crn2	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 2 String OPTIONAL ANS [], MAXLEN=50 Yes Invoice 987
Crn3	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 3 String OPTIONAL ANS [], MAXLEN=50 Yes customer@email.com
Currency	**Description:**	Reserved for future use. Set to null.
MerchantReference	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Merchant reference number String OPTIONAL ANS [], MAXLEN=50 Yes 987654
RedirectionUrl	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Merchant receipt redirection URL String MANDATORY ANS [] Yes https://merchant.com/txnreceipt
WebHookUrl	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Merchant web hook handler URL String OPTIONAL ANS [] Yes https://merchant.com/txnwebhookhandler

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	**Comments:**	
APIResponse	See section 2.3.1	
AuthKey	**Description:**	Unique key that the merchant must use when invoking the “Process Transaction” operation using Browser integration method.. The AuthKey is returned only if the API response code

		indicates success.
	Format:	String
	Constraints:	ANS[all], MAXLEN=500
	Example:	5df3c8cd-f927-4dc0-8ea8-c9a255bee04d

Status Codes:

- 200 OK – Request completed successfully

Request:

```
POST https://www.bpoint.com.au/webapi/v2/txns/processtxnauthkey HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
Content-Length: 278
Expect: 100-continue
Connection: Keep-Alive

{
        "ProcessTxnData" : {
                "Action" : "payment",
                "Amount" : 0,
                "BillerCode" : null,
                "Crn1" : "test crn1",
                "Crn2" : "test crn2",
                "Crn3" : "test crn3",
                "Currency" : "AUD",
                "MerchantReference" : "test merchant ref"
        },
        "RedirectionUrl" : "http:\/\/merchant.com\/txnreceipt",
        "WebHookUrl" : null
}
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 108
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
Set-Cookie: ASP.NET_SessionId=stwcpe0aglzvo5xvdbymbqlg; path=/; HttpOnly
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 02:47:56 GMT

{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        },
        "AuthKey" : "263d58a2-5fd1-46a0-8e83-31c70244f778"
}
```

3.5 Process Transaction using AuthKey (WebForms)

Processes a transaction.

Note: This request is made directly from the card holder's browser. The browser will also automatically handle the response and redirect the card holder to merchant's redirection URL.

POST /txns/webform/process

Integration:

- Browser

URL Parameters:

- None

Request Headers:

- Content-Type: application/x-www-form-urlencoded charset=utf-8

Request Form Data:

Name:	Comments:	
Amount	**Description:**	Transaction amount in the lowest denomination for the currency Ignored if provided in "Create AuthKey" request
	Format:	64 bit integer
	Constraints:	MANDATORY (OPTIONAL if provided in "Create AuthKey" request) MINVAL=1
	Validated:	Yes
	Example:	If AUD, 12000 for $120.00. If JPY, 120 for ¥120
AuthKey	**Description:**	Unique key created using "Create AuthKey" operation
	Format:	String
	Constraints:	MANDATORY ANS[all], MAXLEN=500
	Example:	5df3c8cd-f927-4dc0-8ea8-c9a255bee04d
BillerCode	**Description:**	Biller code. Ignored if provided in "Create AuthKey" request
	Format:	String
	Constraints:	OPTIONAL N, MAXLEN=50
	Validated:	Yes
	Example:	21
CardHolderName	**Description:**	Cardholder name
	Format:	String
	Constraints:	OPTIONAL ANS [all], MAXLEN=50
	Validated:	Yes
	Example:	John Smith
CardNumber	**Description:**	Credit card number
	Format:	String
	Constraints:	MANDATORY N, MINLEN=13, MAXLEN=16
	Validated:	Yes
	Example:	5123456789012346
Crn1	**Description:**	Customer reference number 1 Ignored if provided in "Create AuthKey" request
	Format:	String
	Constraints:	MANDATORY (OPTIONAL if provided in "Create AuthKey" request) ANS [], MAXLEN=50
	Validated:	Yes
	Example:	Customer 1234
Crn2	**Description:**	Customer reference number 2 Ignored if provided in "Create AuthKey" request

	Format: **Constraints:** **Validated:** **Example:**	String OPTIONAL ANS [], MAXLEN=50 Yes Invoice 987
Crn3	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 3 Ignored if provided in “Create AuthKey” request String OPTIONAL ANS [], MAXLEN=50 Yes customer@email.com
Currency	**Description:**	Reserved for future use. Leave blank.
Cvn	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Card verification number String MANDATORY N, MINLEN=3, MAXLEN=4 Yes 123
ExpiryDateMonth	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date month In MM format String MANDATORY N, MINLEN=2, MAXLEN=2 Yes 05
ExpiryDateYear	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date year In YY format String MANDATORY N, MINLEN=2, MAXLEN=2 Yes 17
MerchantReference	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Merchant reference number Ignored if provided in “Create AuthKey” request String OPTIONAL ANS [], MAXLEN=50 Yes 987654
StoreCard	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Flag to indicate whether the cardholder agrees to save their card details. This flag allows merchant to create a DataVault token from the card details used to process the transaction. Boolean OPTIONAL A, if present must be either “true” or “false” Yes true

Response Headers:

- Location: merchant’s redirection URL, API response appended to query string

Response JSON Object:

- None

Status Codes:

- 302 Found – Request completed successfully, redirect to URL found in response “Location” header

Request:

```
POST https://www.bpoint.com.au/webapi/v2/txns/webform/process HTTP/1.1
```

```
Content-Type: application/x-www-form-urlencoded charset=utf-8
Host: www.bpoint.com.au
Content-Length: 263
Expect: 100-continue

&BillerCode=&MerchantReference=test+merchant+ref&Crn1=test+crn1&Crn2=test+crn2&Crn3=test+crn3&Amount=199.
00&CardNumber=5123456789012346&ExpiryDateMonth=05&ExpiryDateYear=17&Cvn=123&CardHolderName=John+S
mith&StoreCard=0&AuthKey=263d58a2-5fd1-46a0-8e83-31c70244f778
```

Response:

```
HTTP/1.1 302 Found
Cache-Control: private
Location: http://merchant.com/txnreceipt?ResponseCode=0&ResponseText=Success&ResultKey=5881fccd-7762-4a4c-
bdc1-cf6ed3b5a3bc
Server: Microsoft-IIS/7.5
Set-Cookie: ASP.NET_SessionId=bq04rkwyfy3hfl50ibljp150; path=/; HttpOnly
p3p: CP="IDC DSP COR ADM DEVi TAIi PSA PSD IVAi IVDi CONi HIS OUR IND CNT"
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 02:47:57 GMT
Content-Length: 0
```

3.6 Process Transaction using AuthKey (JavaScript)

BPOINT API JavaScript makes it easy to process credit card payments without having the information pass through your servers.

Including api.js

```
<script src="https://www.bpoint.com.au/api/cba/api.js?v=2" type="text/javascript"></script>
```

Add this script tag to your page (preferably at the bottom) to get started with the BPOINT JavaScript API. For UAT integration testing please use the url below for the src attribute of script tag - https://bpoint-uat.premier.com.au/api/cba/api.js?v=2

Setting up payment form

```
CBA.SetupPayment({
        AppendToElementId: "pay-form-location",
        AuthKey: $("#AuthKey").val(),
        DefaultErrorUrl: "https://www.yourdomain.com/handleerror"
});
```

This is all you need to start processing payments with BPOINT. The BPOINT JavaScript library will process the payment and redirect the browser to the “RedirectionUrl” supplied when the AuthKey was created (Refer 3.4). At the end of the redirection,

NOTE: Whilst this example uses jQuery's val() to retrieve the values, you can also use standard DOM methods to retrieve card data from your payment form. The BPOINT JavaScript API is JavaScript library agnostic.

invoke the “Retrieve Payment Result” call (Refer 3.7) from your web server to retrieve the payment result and render the payment receipt to your customers.
The three parameters in the sample code above are compulsory. There are a number of optional parameters, which you can pass to the “SetupPayment” method to further customise the payment form as required. Please find below information about the parameters which you can pass to the “SetupPayment” method.

Name:	Comments:	
AppendToElementId	**Description:**	ID of the HTML element on your page where you want the payment form to be inserted.
	Format:	String
	Constraints:	MANDATORY
	Example:	“pay-form-location”
AuthKey	**Description:**	Unique key created using “Create AuthKey” operation (Refer 3.4). You may store this in a hidden field and then pass the value.
	Format:	String
	Constraints:	MANDATORY
	Example:	$("#AuthKey").val()
DefaultErrorUrl	**Description:**	A fall-back URL which the browser is redirected to if a response is not received.
	Format:	String
	Constraints:	MANDATORY
	Example:	“https://www.yourdomain.com/handleerror”
BillerCode	**Description:**	Include this object with appropriate flags if you wish to display the biller code field on the page ǂ
	Format:	Object
	Constraints:	OPTIONAL
	Example:	{ Visible: true, LabelName: "Biller Code", Value: "9999", ReadOnly: false }
Crn1	**Description:**	Include this object with appropriate flags if you wish to display the customer reference number 1 field on the page ǂ
	Format:	Object
	Constraints:	OPTIONAL if supplied while creating the AuthKey (Refer 3.4)
	Example:	{Visible: true, LabelName: "Reference1", Value: "", ReadOnly: false}
Crn2	**Description:**	Include this object with appropriate flags if you wish to display customer reference number 2 field on the page ǂ
	Format:	Object
	Constraints:	OPTIONAL
	Example:	{Visible: true, LabelName: "Reference2", Value: "", ReadOnly: false}
Crn3	**Description:**	Include this object with appropriate flags if you wish to display the

		customer reference number 3 field on the page ǂ
	Format:	Object
	Constraints:	OPTIONAL
	Example:	{Visible: true, LabelName: "Reference3", Value: "", ReadOnly: false}
MerchantReference	**Description:**	Include this object with appropriate flags if you wish to display the merchant reference number field on the page ǂ
	Format:	Object
	Constraints:	OPTIONAL
	Example:	{Visible: true, LabelName: "Extra Reference", Value: "", ReadOnly: false}
Amount	**Description:**	Include this object with appropriate flags if you wish to display amount field on the page ǂ
	Format:	Object
	Constraints:	OPTIONAL
	Example:	{Visible: true, LabelName: "Amount ($)", Value: "10.00", ReadOnly: false}
StoreCard	**Description:**	Include this object with appropriate flags if you wish to offer option for your user to store the card details for future use. This will render a checkbox on the payment form.
	Format:	Object
	Constraints:	OPTIONAL
	Example:	{Visible: true, LabelName: "Remember card", Value: true }

ǂ if this parameter is populated via "Create Auth Key" (Refer 3.4) operation then that value will take precedence over value submitted here.

"SetupPayment" as explained above is the simple approach to begin processing payments using the BPOINT API. For more control over the user experience, please follow the custom payment method as described below.

Process payment

Invoke the process payment method as explained below when you are ready to process the payment e.g. a button click. "ProcessPayment" is an asynchronous call – it returns immediately and invokes the "CallbackFunction" when it receives a response from the BPOINT server.

```
CBA.ProcessPayment({
        AuthKey: $("#AuthKey").val(),
        BillerCode: $("#BillerCode").val(),
        Crn1: $("#CRN1").val(),
        Crn2: $("#CRN2").val(),
        Crn3: $("#CRN3").val(),
        MerchantReference: $("#MerchantReference").val(),
        Amount: $("#Amount").val(),
        CardNumber: $("#CardNumber").val(),
        Cvn: $("#CVN").val(),
        ExpiryMonth: $("#ExpMonth").val(),
        ExpiryYear: $("#ExpYear").val(),
        CardHolderName: $("#CardHolderName").val(),
        StoreCard: $("#StoreCard").prop("checked"),
        CallbackFunction: ProcessPaymentCallBack
});
```

Please find below information on the parameters which you can pass in to the “ProcessPayment” method.

Name:	Comments:	
AuthKey	**Description:**	Unique key created using “Create AuthKey” operation (Refer 3.4). You may store this in a hidden field and then pass the value.
	Format:	String
	Constraints:	MANDATORY
BillerCode	**Description:**	Biller code value to attach to a payment
	Format:	String
	Constraints:	OPTIONAL
Crn1	**Description:**	Customer reference number 1 for the payment
	Format:	String
	Constraints:	OPTIONAL if supplied while creating the AuthKey (Refer 3.4)
Crn2	**Description:**	Customer reference number 2 for the payment
	Format:	String
	Constraints:	OPTIONAL
Crn3	**Description:**	Customer reference number 3 for the payment
	Format:	String
	Constraints:	OPTIONAL
MerchantReference	**Description:**	Merchant Reference for the payment
	Format:	String
	Constraints:	OPTIONAL
Amount	**Description:**	Amount to process
	Format:	String
	Constraints:	OPTIONAL if supplied while creating the AuthKey (Refer 3.4)
CardNumber	**Description:**	Card number you wish to charge
	Format:	Numeric
	Constraints:	MANDATORY
Cvn	**Description:**	Card verification number
	Format:	Numeric
	Constraints:	MANDATORY
ExpiryMonth	**Description:**	Expiry month of the card
	Format:	Numeric

	Constraints:	MANDATORY
ExpiryYear	**Description:**	Expiry year of the card
	Format:	Numeric
	Constraints:	MANDATORY
CardHolderName	**Description:**	Name on the card
	Format:	String
	Constraints:	OPTIONAL
StoreCard	**Description:**	Set this flag if you wish to create a DataVault to store the card details for future use. Alternatively you may render a checkbox on your form for your customer to opt in.
	Format:	Boolean (true or false)
	Constraints:	OPTIONAL (default is false)
CallbackFunction	**Description:**	This is a callback you provide to handle the response from the BPOINT API. Please see below for more information on this.
	Format:	Function
	Constraints:	MANDATORY

The “CallbackFunction” is a JavaScript function you provide to handle the response from the BPOINT API. It does the following:

1. If the payment information returned an error, display it on the page; or
2. If an error was not returned, then submit the result back to your server to call the “Retrieve Transaction Result” (Refer 3.7) and render the receipt. If you are integrating for a shopping cart then this would be the time to check out the cart. **Note:** It is recommended that you do not submit the card details to your server, and only submit the result of the call back to your server.

Below is a sample implementation of the “ProcessPaymentCallBack”:

```
function ProcessPaymentCallBack(result) {
        var errors = new Array();

        if (result.AjaxResponseType == 0) { //AJAX call was successful

                if (result.ApiResponseCode == 0) {
                //API returned success. Refer to Appendix 6.1 for API Response codes.
                //Submit result.ResultKey to your server for further processing (Refer 3.7)
                }
                else {
                        errors = result.Errors;
                }
        }
        else if(result.AjaxResponseType == 1) { //Error with AJAX call
                errors = result.Errors;
        }
```

```
            else if(result.AjaxResponseType == 2) { //AJAX call timed out
                        errors = result.Errors;
            }

            //Show errors on the page
            if (errors.length > 0) {
                        var ul = $("<ul></ul>");

                        $.each(errors, function (i, r) {
                                    ul.append("<li>" + r.Message + "</li>");
                        });

                        $(".validation-summary").append(ul).show();
            }
}
```

The result object is described as below:

Name:	Comments:	
AjaxResponseType	**Description:**	This is the result of the AJAX call. This will assist you in handling the call back.
	Format:	Numeric
	Values:	Possible values are 0, 1 or 2. Where 0 is success, 1 is error and 2 is timeout.
ApiResponseCode	**Description:**	This is the response code returned by the BPOINT API
	Format:	Numeric
	Values:	Please refer to Appendix 6.1 for possible values
Errors	**Description:**	List of errors returned e.g. validation errors.
	Format:	Array of object {Message: “Invalid card number”,PropertyName: “CardNumber”}
ResultKey	**Description:**	This is the result key that you will have to submit to retrieve the transaction result and present the receipt to the customer,“Retrieve Transaction Result” (Refer 3.7).
	Format:	String

3.7 Retrieve Transaction Result

Retrieves the result of a transaction processed via an operation using the Browser integration method.

GET /txns/withauthkey/{resultKey}

Integration:

- Direct

URL Parameters:

- resultKey – result key returned in a response to process transaction operation

Request Headers:

- Authorisation: base64 encoded API user credentials

Request JSON Object:

- None

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
TxnResp	See section 2.4.3.5

Status Codes:

- 200 OK – Request completed successfully

Request:

```
GET http://www.bpoint.com.au/api/v2/txns/withauthkey/5881fccd-7762-4a4c-bdc1-cf6ed3b5a3bc HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Host: www.bpoint.com.au
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 899
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 02:47:57 GMT

{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        },
        "TxnResp" : {
                "Action" : "payment",
                "Amount" : 19900,
                "AmountSurcharge" : 0,
                "AmountSurchargeGST" : 0,
                "AuthoriseId" : "634831",
                "BankAccountDetails" : null,
                "BankResponseCode" : "00",
                "BillerCode" : "",
                "CVNResult" : {
                        "CVNResultCode" : "Unsupported"
                },
                "CardDetails" : {
                        "CardHolderName" : "John Smith",
                        "ExpiryDate" : "0517",
                        "MaskedCardNumber" : "512345...346"
                },
                "CardType" : "MC",
                "Crn1" : "test crn1",
                "Crn2" : "test crn2",
                "Crn3" : "test crn3",
                "Currency" : null,
                "IsCVNPresent" : true,
                "IsThreeDS" : false,
                "MerchantNumber" : "0000000000000000",
                "MerchantReference" : "test merchant ref",
                "OriginalTxnNumber" : null,
                "ProcessedDateTime" : "2014-11-05T16:47:57.4730000",
                "RRN" : "431013634831",
                "ReceiptNumber" : "48346146587",
                "ResponseCode" : "0",
                "ResponseText" : "Approved",
                "SettlementDate" : "20141106",
                "Source" : "api",
                "StoreCard" : false,
                "SubType" : "single",
                "ThreeDSResponse" : null,
                "TxnNumber" : "46476587",
                "Type" : "internet"
        }
}
```

4 DataVault Tokens

4.1 Add DataVault Token

Securely stores payment details and creates a unique DataVault token.

POST /dvtokens

Integration:

- Direct

URL Parameters:

- None

Request Headers:

- Authorisation: base64 encoded API user credentials
- Content-Type: application/json; charset=utf-8

Request JSON Object:

Name:	Comments:
DVTokenReq	See section 2.4.2.4

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
DVTokenResp	See section 2.4.3.6

Status Codes:

- 200 OK – Request completed successfully

Request:

```
POST https://www.bpoint.com.au/webapi/v2/dvtokens/ HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
Content-Length: 259
Expect: 100-continue
Connection: Keep-Alive

{
        "DVTokenReq" : {
                "AcceptBADirectDebitTC" : false,
                "BankAccountDetails" : null,
                "CardDetails" : {
                        "CardHolderName" : "John Smith",
                        "CardNumber" : "5123456789012346",
                        "Cvc" : "123",
                        "ExpiryDate" : "0517"
                },
                "Crn1" : "12345",
                "Crn2" : "",
                "Crn3" : null,
                "EmailAddress" : "john.smith@email.com.au"
        }
}
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 321
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
Set-Cookie: ASP.NET_SessionId=oxcbbbamkjcoht1ue0isxclc; path=/; HttpOnly
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 03:14:05 GMT

{
          "APIResponse" : {
                    "ResponseCode" : 0,
                    "ResponseText" : "Success"
          },
          "DVTokenResp" : {
                    "BankAccountDetails" : null,
                    "CardDetails" : {
                              "CardHolderName" : "John Smith",
                              "ExpiryDate" : "0517",
                              "MaskedCardNumber" : "512345...346"
                    },
                    "CardType" : "MC",
                    "Crn1" : "12345",
                    "Crn2" : "",
                    "Crn3" : null,
                    "EmailAddress" : "john.smith@email.com.au",
                    "DVToken" : "5999991550628061"
          }
}
```

4.2 Update a DataVault Token

Updates details for a given DataVault token.

PUT /dvtokens/{dvtoken}

Integration:

- Direct

URL Parameters:

- dvtoken – DataVault token to be updated

Request Headers:

- Authorisation: base64 encoded API user credentials
- Content-Type: application/json; charset=utf-8

Request JSON Object:

Name:	Comments:
DVTokenReq	See section 2.4.2.4

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
DVTokenResp	See section 2.4.3.6

Status Codes:

- 200 OK – Request completed successfully

Request:

```
PUT https://www.bpoint.com.au/webapi/v2/dvtokens/5999991550628160 HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
Content-Length: 239
Expect: 100-continue

{
	"DVTokenReq" : {
		"AcceptBADirectDebitTC" : true,
		"BankAccountDetails" : {
			"AccountName" : "Tommy Smith",
			"AccountNumber" : "123123",
			"BSBNumber" : "062000"
		},
		"CardDetails" : null,
		"Crn1" : "12345",
		"Crn2" : "",
		"Crn3" : null,
		"EmailAddress" : "tommy.smith@email.com.au"
	}
}
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 303
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 03:25:56 GMT

{
	"APIResponse" : {
		"ResponseCode" : 0,
		"ResponseText" : "Success"
	},
	"DVTokenResp" : {
		"BankAccountDetails" : {
```

```
                              "AccountName" : "Tommy Smith",
              "AccountNumber" : "123123",
                              "BSBNumber" : "062000",
                              "TruncatedAccountNumber" : "062000###123"
                    },
                    "CardDetails" : null,
                    "CardType" : "BA",
                    "Crn1" : "12345",
                    "Crn2" : "",
                    "Crn3" : "",
                    "EmailAddress" : "tommy.smith@email.com.au",
                    "DVToken" : "5999991550628160"
          }
}
```

4.3 Tokenise a Card from a Transaction

Creates a DataVault token using payment details from a previously processed transaction.

POST /dvtokens/txn/{txnNumber}

Integration:

- Direct

URL Parameters:

- txnNumber - transaction number of a previously processed transaction

Request Headers:

- Authorisation: base64 encoded API user credentials
- Content-Type: application/json; charset=utf-8

Request JSON Object:

- None

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
DVTokenResp	See section 2.4.3.6

Status Codes:

- 200 OK – Request completed successfully

Request:

```
POST https://www.bpoint.com.au/webapi/v2/dvtokens/txn/46476588 HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
Content-Length: 0
```

Response:

```
HTTP/1.1 200 OK
```

```
Cache-Control: private
Content-Length: 311
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
Set-Cookie: ASP.NET_SessionId=de3btvjwgaqurwq0zsx2hxpf; path=/; HttpOnly
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 03:29:38 GMT

{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        },
        "DVTokenResp" : {
                "BankAccountDetails" : null,
                "CardDetails" : {
                        "CardHolderName" : null,
                        "ExpiryDate" : "9900",
                        "MaskedCardNumber" : "512345...346"
                },
                "CardType" : "MC",
                "Crn1" : "test crn1",
                "Crn2" : "test crn2",
                "Crn3" : "test crn3",
                "EmailAddress" : null,
                "DVToken" : "5999991550628269"
        }
}
```

4.4 Retrieve a DataVault Token

Retrieves DataVault token details.

GET /dvtokens/{dvtoken}

Integration:

- Direct

URL Parameters:

- dvtoken – token record to retrieve

Request Headers:

- Authorisation: base64 encoded API user credentials

Request JSON Object:

- None

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
DVTokenResp	See section 2.4.3.6

Status Codes:

- 200 OK – Request completed successfully

Request:

```
GET https://www.bpoint.com.au/webapi/v2/dvtokens/5999991550628368 HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Host: www.bpoint.com.au
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 319
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 03:34:46 GMT

{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        },
        "DVTokenResp" : {
                "BankAccountDetails" : null,
                "CardDetails" : {
                        "CardHolderName" : "John Smith",
                        "ExpiryDate" : "0517",
                        "MaskedCardNumber" : "512345...346"
                },
                "CardType" : "MC",
                "Crn1" : "12345",
                "Crn2" : "",
                "Crn3" : "",
                "EmailAddress" : "john.smith@email.com.au",
                "DVToken" : "5999991550628368"
        }
}
```

4.5 Search DataVault Tokens

Performs a search on the merchant’s DataVault tokens.

POST /dvtokens/search

Integration:

- Direct

URL Parameters:

- None

Request Headers:

- Authorisation: base64 encoded API user credentials

- Content-Type: application/json; charset=utf-8

Request JSON Object:

Name:	Comments:	
SearchInput	**Description:**	JSON object containing search details.
CardType	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Card type String OPTIONAL A, MAXLEN=2 No MC
Crn1	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 1 String OPTIONAL ANS [], MAXLEN=50 No Customer 1234
Crn2	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 2 String OPTIONAL ANS [], MAXLEN=50 No Invoice 987
Crn3	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 3 String OPTIONAL ANS [], MAXLEN=50 No customer@email.com
ExpiredCardsOnly	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Flag to indicate search should only return DataVault tokens where the card expiry date is in the past. Boolean OPTIONAL A, if present must be either “true” or “false” No true
ExpiryDate	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date In MMYY format String OPTIONAL N, MINLEN=4, MAXLEN=4 No 0517
FromDate	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Start date for a date range search, compared against the token creation date. In ISO8601 format. String OPTIONAL ANS [], MAXLEN=50 No 2014-11-05T12:53:27.4738695+11:00
MaskedCardNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Masked card number. Can also be masked bank account details if searching for bank account tokens, eg: 061200###123 String OPTIONAL ANS [.#], MAXLEN=12 No 512345…346
Source	**Description:** **Format:** **Constraints:**	Transaction origin. See section 6.6 String OPTIONAL A, MAXLEN=50

	Validated: **Example:**	No api
ToDate	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	End date for a date range search, compared against the token creation date. In ISO8601 format. String OPTIONAL ANS [], MAXLEN=50 No 2014-11-05T12:53:27.4738695+11:00
DVToken	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	DataVault token String OPTIONAL N, MAXLEN=16 No 5999991550629366
UserCreated	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Username of the user who created the DataVault token String OPTIONAL ANS [], MAXLEN=50 No apiuser
UserUpdated	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Username of the user who last updated the DataVault token String OPTIONAL ANS [], MAXLEN=50 No apiuser

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
DVTokenRespList	Array of token response objects. See section 2.4.3.6

Status Codes:

- 200 OK – Request completed successfully

Request:

```
POST https://www.bpoint.com.au/webapi/v2/dvtokens/search HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
Content-Length: 293
Expect: 100-continue
Connection: Keep-Alive

{
        "SearchInput" : {
                "CardType" : null,
                "Crn1" : null,
                "Crn2" : null,
                "Crn3" : null,
                "ExpiredCardsOnly" : false,
                "ExpiryDate" : null,
                "FromDate" : "2014-11-05T15:02:49.9954695+11:00",
                "MaskedCardNumber" : null,
                "Source" : null,
                "ToDate" : "2014-11-06T15:02:49.9954695+11:00",
                "DVToken" : null,
                "UserCreated" : null,
                "UserUpdated" : null
        }
}
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 3172
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
Set-Cookie: ASP.NET_SessionId=iqmtjpq230veivvtrx1nal3d; path=/; HttpOnly
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 04:03:11 GMT

{
          "APIResponse" : {
                    "ResponseCode" : 0,
                    "ResponseText" : "Success"
          },
          "DVTokenRespList" : [{
                              "BankAccountDetails" : {
                                        "AccountName" : "Jimmy",
                  "AccountNumber" : "123678",
                              "BSBNumber" : "063001"
                                        "TruncatedAccountNumber" : "063001###678"
                              },
                              "CardDetails" : null,
                              "CardType" : "BA",
                              "Crn1" : "1234567",
                              "Crn2" : "",
                              "Crn3" : "",
                              "EmailAddress" : "jimmy@email.com.au",
                              "DVToken" : "5999991550627287"
                    }, ..., {
                              "BankAccountDetails" : null,
                              "CardDetails" : {
                                        "CardHolderName" : "Test123456",
                                        "ExpiryDate" : "0519",
                                        "MaskedCardNumber" : "512345...346"
                              },
                              "CardType" : "MC",
                              "Crn1" : "1234567",
                              "Crn2" : "",
                              "Crn3" : "",
                              "EmailAddress" : "",
                              "DVToken" : "5999991550628467"
                    }
          ]
}
```

4.6 Delete DataVault Token

Deletes an existing DataVault token.

DELETE /dvtokens/{dvtoken}

Integration:

- Direct

URL Parameters:

- dvtoken – token record to retrieve

Request Headers:

- Authorisation: base64 encoded API user credentials

Request JSON Object:

- None

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1

Status Codes:

- 200 OK – Request completed successfully

Request:

```
DELETE https://www.bpoint.com.au/webapi/v2/dvtokens/5999991550628566 HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
Content-Length: 0
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 59
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 04:05:50 GMT

{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        }
}
```

4.7 Create AuthKey To Add a DataVault Token

Creates one-time AuthKey to be used with "Add DataVault Token" operation when using **Browser** integration method.

This operation serves following functions:

- Authenticates the merchant by username, merchant number and password. If these details are not correct, the AuthKey will not be generated.
- Creates a unique AuthKey to allow merchant's customer to save their payment details.
- AuthKey prevents processing of duplicate requests.
- Allows merchant to "lock in" certain details and means they cannot be modified even if submitted in subsequent "Add DataVault Token" request. These fields include:
    - Crn1
    - Crn2
    - Crn3
    - EmailAddress
- AuthKey is valid only for a predefined period of time (20 minutes). If the request is not attempted during that time merchant's application will need to request a new AuthKey.

POST /dvtokens/adddvtokenauthkey

Integration:

- Direct

URL Parameters:

- None

Request Headers:

- Authorisation: base64 encoded API user credentials
- Content-Type: application/json; charset=utf-8

Request JSON Object:

Name:	Comments:
FixedAddDVTokenData	
Crn1	**Description:** Customer reference number 1 **Format:** String **Constraints:** OPTIONAL ANS [], MAXLEN=50 **Validated:** Yes **Example:** Customer 1234
Crn2	**Description:** Customer reference number 2 **Format:** String **Constraints:** OPTIONAL ANS [], MAXLEN=50 **Validated:** Yes **Example:** Invoice 987
Crn3	**Description:** Customer reference number 3 **Format:** String **Constraints:** OPTIONAL ANS [], MAXLEN=50 **Validated:** Yes **Example:** customer@email.com
EmailAddress	**Description:** Email address **Format:** String **Constraints:** OPTIONAL ANS [], MAXLEN=250 **Validated:** Yes **Example:** customer@email.com

RedirectionUrl	**Description:** Merchant redirection URL **Format:** String **Constraints:** MANDATORY ANS [] **Validated:** Yes **Example:** https://merchant.com/tokenreceipt
WebHookUrl	**Description:** Merchant web hook handler URL **Format:** String **Constraints:** OPTIONAL ANS [] **Validated:** Yes **Example:** https://merchant.com/tokenwebhookhandler

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
AuthKey	**Description:** Unique key that the merchant must use when invoking the "Add DataVault Token" operation using the Browser integration method. The AuthKey is returned only if the API response code indicates success. **Format:** String **Constraints:** ANS[all], MAXLEN=500 **Example:** 5df3c8cd-f927-4dc0-8ea8-c9a255bee04d

Status Codes:

- 200 OK – Request completed successfully

Request:

```
POST https://www.bpoint.com.au/webapi/v2/dvtokens/adddvtokenauthkey HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
Content-Length: 91
Expect: 100-continue

{
        "FixedAddDVTokenData" : null,
        "RedirectionUrl" : "http:\/\/merchant.com\/tokenreceipt",
        "WebHookUrl" : null
}
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 108
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
Set-Cookie: ASP.NET_SessionId=e55eydlrdxhnl2gmh4lyyrsd; path=/; HttpOnly
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
```

```
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 21:42:46 GMT

{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        },
        "AuthKey" : "6158ab0a-ea1b-4402-927a-38f17a1d5dde"
}
```

4.8 Create an AuthKey to Update a DataVault Token

Creates a one-time AuthKey to be used with the “Update DataVaultToken” operation when using the **Browser** integration method.

This operation serves the following functions:

- Authenticates the merchant by username, merchant number and password. If these details are not correct, the AuthKey will not be generated.
- Creates a unique AuthKey to allow the merchant’s customer to update their payment details.
- AuthKey prevents processing of duplicate requests.
- Allows merchant to “lock in” certain details and prevent the details from being modified. These fields include:
    - Crn1
    - Crn2
    - Crn3

- AuthKey is valid only for a predefined period of time (20 minutes). If the request is not attempted within that time, the merchant’s application will need to request a new AuthKey.

POST /dvtokens/updatedvtokenauthkey

Integration:

- Direct

URL Parameters:

- None

Request Headers:

- Authorisation: base64 encoded API user credentials
- Content-Type: application/json; charset=utf-8

Request JSON Object:

Name:	Comments:
FixedUpdateDVTokenData	
Crn1	**Description:** Customer reference number 1 **Format:** String **Constraints:** OPTIONAL ANS [], MAXLEN=50 **Validated:** Yes **Example:** Customer 1234
Crn2	**Description:** Customer reference number 2 **Format:** String

	Constraints: **Validated:** **Example:**	OPTIONAL ANS [], MAXLEN=50 Yes Invoice 987
Crn3	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 3 String OPTIONAL ANS [], MAXLEN=50 Yes customer@email.com
EmailAddress	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Email address String OPTIONAL ANS [], MAXLEN=250 Yes customer@email.com
DVToken	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	DataVault token to update String MANDATORY N, MAXLEN=16 Yes 5999991550629366
RedirectionUrl	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Merchant redirection URL String MANDATORY ANS [] Yes https://merchant.com/tokenreceipt
WebHookUrl	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Merchant web hook handler URL String OPTIONAL ANS [] Yes https://merchant.com/tokenwebhookhandler

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	**Comments:**	
APIResponse	See section 2.3.1	
AuthKey	**Description:** **Format:** **Constraints:** **Example:**	Unique key that the merchant must use when invoking the “Update DataVault Token” operation using the Browser integration method.. The AuthKey is returned only if the API response code indicates success. String ANS[all], MAXLEN=500 5df3c8cd-f927-4dc0-8ea8-c9a255bee04d

Status Codes:

- 200 OK – Request completed successfully

Request:

```
POST https://www.bpoint.com.au/webapi/v2/dvtokens/updatedvtokenauthkey HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Content-Type: application/json; charset=utf-8
Host: www.bpoint.com.au
```

```
Content-Length: 242
Expect: 100-continue

{
        "FixedUpdateDVTokenData" : {
                "Crn1" : "test crn1",
                "Crn2" : "test crn2",
                "Crn3" : "test crn3",
                "EmailAddress" : "john.smith@email.com.au",
                "DVToken" : "5999991550629366"
        },
        "RedirectionUrl" : "http:\/\/merchant.com\/tokenreceipt",
        "WebHookUrl" : null
}
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 108
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 21:53:00 GMT

{
        "APIResponse" : {
                "ResponseCode" : 0,
                "ResponseText" : "Success"
        },
        "AuthKey" : "8eeacd3c-68b0-42c4-8566-da12e83c5d74"
}
```

4.9 Add a DataVault Token using an AuthKey (WebForms)

Securely stores payment details and creates a unique DataVault token.

Note: This request is made directly from the cardholder’s browser. The browser will also automatically handle the response and redirect the cardholder to the appropriate URL.

POST /dvtokens/webform/add

Integration:

- Browser

URL Parameters:

- None

Request Headers:

- Content-Type: application/x-www-form-urlencoded charset=utf-8

Request Form Data:

Name:	Comments:	
AuthKey	**Description:** **Format:** **Constraints:** **Example:**	Unique key created using the “Create AuthKey” operation String MANDATORY ANS[all], MAXLEN=500 5df3c8cd-f927-4dc0-8ea8-c9a255bee04d
CardHolderName	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Cardholder name String OPTIONAL ANS [all], MAXLEN=50 Yes John Smith
CardNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card number String MANDATORY N, MINLEN=13, MAXLEN=16 Yes 5123456789012346
Cvn	**Not required and will be ignored if included in the request. CVN fields are never stored in BPOINT.**	
ExpiryDateMonth	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date month In MM format String MANDATORY N, MINLEN=2, MAXLEN=2 Yes 05
ExpiryDateYear	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date year In YY format String MANDATORY N, MINLEN=2, MAXLEN=2 Yes 17
AccountName	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Bank account name String MANDATORY ANS [EBCDIC character set], MINLEN=1, MAXLEN=32 Yes John Smith
AccountNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Bank account number String MANDATORY AN, MINLEN=3, MAXLEN=9 Yes 00123
BSBNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	BSB number String MANDATORY N, MINLEN=6, MAXLEN=6 Yes 061200
Crn1	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 1 Ignored if provided in “Create AuthKey” request String MANDATORY (OPTIONAL if provided in “Create AuthKey” request) ANS [], MAXLEN=50 Yes Customer 1234
Crn2	**Description:**	Customer reference number 2 Ignored if provided in “Create AuthKey” request

	Format: **Constraints:** **Validated:** **Example:**	String OPTIONAL ANS [], MAXLEN=50 Yes Invoice 987
Crn3	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 3 Ignored if provided in “Create AuthKey” request String OPTIONAL ANS [], MAXLEN=50 Yes customer@email.com
EmailAddress	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Email address Ignored if provided in “Create AuthKey” request String OPTIONAL ANS [], MAXLEN=250 Yes customer@email.com

Response Headers:

- Location: URL to redirect to

Response JSON Object:

- None

Status Codes:

- 302 Found – Request completed successfully, redirect to URL found in response “Location” header

Request:

```
POST https://www.bpoint.com.au/webapi/v2/dvtokens/webform/add HTTP/1.1
Content-Type: application/x-www-form-urlencoded charset=utf-8
Host: www.bpoint.com.au
Content-Length: 205
Expect: 100-continue

&Crn1=12345&Crn2=&Crn3=&EmailAddress=cardholder%40email.com.au&CardNumber=5123456789012346&ExpiryD
ateMonth=05&ExpiryDateYear=17&Cvn=&CardHolderName=John+Smith&AuthKey=6158ab0a-ea1b-4402-927a-
38f17a1d5dde
```

Response:

```
HTTP/1.1 302 Found
Cache-Control: private
Location: http://merchant.com/tokenreceipt?ResponseCode=0&ResponseText=Success&ResultKey=208e4a07-ac75-
479e-ac4e-66ada2066f00
Server: Microsoft-IIS/7.5
Set-Cookie: ASP.NET_SessionId=dgnhacbnkqso5rhzaeflhpvx; path=/; HttpOnly
p3p: CP="IDC DSP COR ADM DEVi TAIi PSA PSD IVAi IVDi CONi HIS OUR IND CNT"
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 21:42:46 GMT
Content-Length: 0
```

4.10 Add a DataVault Token using a AuthKey (JavaScript)

BPOINT API JavaScript makes it easy to store credit card details without having the card data pass through your servers.

Including api.js

```
<script src="https://www.bpoint.com.au/api/cba/api.js?v=2" type="text/javascript"></script>
```

Add this script tag to your page (preferably at the bottom) to get started with BPOINT JavaScript API. For UAT integration testing please use the URL below for the src attribute of script tag - https://bpoint-uat.premier.com.au/api/cba/api.js?v=2

Setting up add DataVault token form

```
CBA.SetupAddToken({
            AppendToElementId: "add-form-location",
            AuthKey: $("#AuthKey").val(),
            DefaultErrorUrl: "https://www.yourdomain.com/handleerror"
});
```

This is all you need to start tokenising cards with BPOINT. The BPOINT JavaScript library will store the card details and redirect the browser to the "RedirectionUrl" supplied when the AuthKey was created (Refer 4.7). At the end of the redirection, invoke the "Lookup Add/Update DataVault Token Result" call (Refer 4.13) from your web server to retrieve the DataVault token information and render the confirmation receipt to your customers.

NOTE: Whilst this example, uses jQuery's val() to retrieve values, you can also use standard DOM methods to retrieve card data from your form. BPOINT JavaScript API is JavaScript library agnostic.

The three parameters in the sample code above are compulsory. There are a number of optional parameters, which you can pass to the "SetupAddToken" method to further customise the form as required. Please find below information about the parameters which you can pass to "SetupAddToken" method.

Name:	Comments:	
AppendToElementId	**Description:**	ID of the HTML element on your page where you want the form to be inserted.
	Format:	String
	Constraints:	MANDATORY
	Example:	"add-form-location"
AuthKey	**Description:**	Unique key created using "Create AuthKey" operation (Refer 4.7). You may store this in a hidden field and then pass the value.
	Format:	String
	Constraints:	MANDATORY
	Example:	$("#AuthKey").val()
DefaultErrorUrl	**Description:**	A fall-back URL which the browser is redirected to if a response is

	not received. **Format:** String **Constraints:** MANDATORY **Example:** “https://www.yourdomain.com/handleerror”
Crn1	**Description:** Include this object with appropriate flags if you wish to display customer reference number 1 field on the page [ǂ] **Format:** Object **Constraints:** OPTIONAL if supplied while creating the AuthKey (Refer 4.7) **Example:** {Visible: true, LabelName: "Reference1", Value: "", ReadOnly: false}
Crn2	**Description:** Include this object with appropriate flags if you wish to display customer reference number 2 field on the page [ǂ] **Format:** Object **Constraints:** OPTIONAL **Example:** {Visible: true, LabelName: "Reference2", Value: "", ReadOnly: false}
Crn3	**Description:** Include this object with appropriate flags if you wish to display customer reference number 3 field on the page [ǂ] **Format:** Object **Constraints:** OPTIONAL **Example:** {Visible: true, LabelName: "Reference3", Value: "", ReadOnly: false}
EmailAddress	**Description:** Include this object with appropriate flags if you wish to display email address field on the page [ǂ] **Format:** Object **Constraints:** OPTIONAL **Example:** {Visible: true, LabelName: "Email", Value: "", ReadOnly: false}
Type	**Description:** The type of information that you wish to store. E.g. Credit card or bank account. You may also choose to offer this choice to your customer. **Format:** String **Constraints:** OPTIONAL **Example:** CREDITCARD - this will render only credit card fields on the form BANKACCOUNT – this will render only bank account fields on the form OPTION – This will render the form with radio button, offering the user the option to choose which type of information they want to store.
AcceptBankAccountTerms	**Description:** This must be true for the Bank account token type. This notifies BPOINT server that you have rendered appropriate terms and conditions to the user and the user has accepted. This flag can be ignored for the credit card token type. **Format:** Boolean **Constraints:** MANDATORY if storing Bank account tokens, optional otherwise.

	Example: true

ǂ if this parameter is populated via “Create Auth Key” (Refer 4.7) operation then that value will take precedence over value submitted here.

“SetupAddToken” as explained above is the simple approach to start collecting credit card or bank account information using the BPOINT API. For more control over the user experience, please follow the custom method as described below.

Process add DataVault token

Invoke the process add DataVault token method as explained below when you are ready to store the payment instrument e.g. a button click. “ProcessAddToken” is an asynchronous call – it returns immediately and invokes the “CallbackFunction” when it receives a response from the BPOINT server.

```
CBA.ProcessAddToken({
        AuthKey: $("#AuthKey").val(),
        Crn1: $("#CRN1").val(),
        Crn2: $("#CRN2").val(),
        Crn3: $("#CRN3").val(),
        EmailAddress: $("#EmailAddress").val(),
        Type: “CREDITCARD”, //Either CREDITCARD or BANKACCOUNT
        CardHolderName: $("#CardHolderName").val(),
        CardNumber: $("#CardNumber").val(),
        ExpiryMonth: $("#CardExpiryMonth").val(),
        ExpiryYear: $("#CardExpiryYear").val(),
        BSBNumber: $("#BsbNumber").val(),
        BankAccountNumber: $("#BankAccountNumber").val(),
        BankAccountName: $("#BankAccountName").val(),
        AcceptBankAccountTerms: $("#AcceptBankAccountTerms").prop("checked"),
        CallbackFunction: ProcessAddTokenCallBack
});
```

Please find below information on the parameters which you can pass in to “ProcessAddToken” method.

Name:	Comments:
AuthKey	**Description:** Unique key created using the “Create AuthKey” operation (Refer 4.7). You may store this in a hidden field and then pass the value. **Format:** String **Constraints:** MANDATORY
Crn1	**Description:** Customer reference number 1 to store with the DataVault token **Format:** String **Constraints:** OPTIONAL if supplied while creating the AuthKey (Refer 4.7)
Crn2	**Description:** Customer reference number 2 to store with the DataVault token **Format:** String **Constraints:** OPTIONAL

Crn3	**Description:**	Customer reference number 3 to store with the DataVault token
	Format:	String
	Constraints:	OPTIONAL
EmailAddress	**Description:**	Email address to store with the DataVault token
	Format:	String
	Constraints:	OPTIONAL
Type	**Description:**	The type of information that you wish to store. E.g. Credit card or bank account. You may also choose to offer this choice to your customer via radio buttons.
	Format:	String
	Constraints:	OPTIONAL (Default is CREDITCARD)
	Value	Either CREDITCARD or BANKACCOUNT
CardHolderName	**Description:**	Name on the card
	Format:	String
	Constraints:	OPTIONAL
CardNumber	**Description:**	Card number you wish to tokenise
	Format:	Numeric
	Constraints:	MANDATORY if TYPE is CREDITCARD
ExpiryMonth	**Description:**	Expiry month of the card
	Format:	Numeric
	Constraints:	MANDATORY if TYPE is CREDITCARD
ExpiryYear	**Description:**	Expiry year of the card
	Format:	Numeric
	Constraints:	MANDATORY if TYPE is CREDITCARD
BSBNumber	**Description:**	BSB number you wish to store
	Format:	String
	Constraints:	MANDATORY if TYPE is BANKACCOUNT
BankAccountNumber	**Description:**	Bank account number you wish to store
	Format:	String
	Constraints:	MANDATORY if TYPE is BANKACCOUNT
BankAccountName	**Description:**	Bank account name
	Format:	String
	Constraints:	MANDATORY if TYPE is BANKACCOUNT
AcceptBankAccountTerms	**Description:**	This must be true for the Bank account token type. This notifies BPOINT server that you have rendered appropriate terms and conditions to the user and the user has accepted. This flag can be ignored for the credit card token type.
	Format:	Boolean (true or false)
	Constraints:	MANDATORY if TYPE is BANKACCOUNT
CallbackFunction	**Description:**	This is a callback you provide to handle the response from the BPOINT API. Please see below for more information on this.

	Format: Function **Constraints:** MANDATORY

The “CallbackFunction” is a JavaScript function you provide to handle the response from the BPOINT API. It does the following

1. If the payment information entered by the user returned an error, display it on the page
2. If an error was not returned, then submit the result back to your server to call the “Lookup Add/Update DataVault Token Result” (Refer 4.13) and render the conformation receipt.
   Note: It is recommended that you do not submit the card details to your server, and only submit the result of the call back to your server.

Below is a sample implementation of “ProcessAddTokenCallBack”:

```
function ProcessAddTokenCallBack(result) {
        var errors = new Array();

        if (result.AjaxResponseType == 0) { //AJAX call was successful

                if (result.ApiResponseCode == 0) {
                //API returned success. Refer to Appendix 6.1 for API Response codes.
                //Submit result.ResultKey to your server for further processing(Refer 4.13)
                }
                else {
                        errors = result.Errors;
                }
        }
        else if(result.AjaxResponseType == 1) { //Error with AJAX call
                errors = result.Errors;
        }
        else if(result.AjaxResponseType == 2) { //AJAX call timed out
                errors = result.Errors;
        }

        //Show errors on the page
        if (errors.length > 0) {
                var ul = $("<ul></ul>");

                $.each(errors, function (i, r) {
                        ul.append("<li>" + r.Message + "</li>");
                });

                $(".validation-summary").append(ul).show();
        }
}
```

The result object is described as below:

Name:	Comments:
AjaxResponseType	**Description:** This is the result of the AJAX call. This will assist you in handling the call back. **Format:** Numeric **Values:** Possible values are 0, 1 or 2. Where 0 is success, 1 is error and 2 is timeout.
ApiResponseCode	**Description:** This is the response code returned by the BPOINT API

	Format: Numeric **Values:** Please refer to Appendix 6.1 for possible values
Errors	**Description:** List of errors returned, e.g. validation errors. **Format:** Array of object {Message: "Invalid card number",PropertyName: "CardNumber"}
ResultKey	**Description:** This is the result key that you will have to submit to retrieve the DataVault token information, "Lookup Add/Update DataVault Token Result" (Refer 4.13). **Format:** String

4.11 Update DataVault Token using AuthKey (WebForms)

Updates payment details for existing an existing DataVault token.

Note: This request is made directly from the cardholder's browser. The browser will also automatically handle the response and redirect the cardholder to the appropriate URL.

POST /dvtokens/webform/update

Integration:

- Browser

URL Parameters:

- None

Request Headers:

- Content-Type: application/x-www-form-urlencoded charset=utf-8

Request Form Data:

Name:	Comments:
AuthKey	**Description:** Unique key created using "Create AuthKey" operation **Format:** String **Constraints:** MANDATORY ANS[all], MAXLEN=500 **Example:** 5df3c8cd-f927-4dc0-8ea8-c9a255bee04d
CardHolderName	**Description:** Cardholder name **Format:** String **Constraints:** OPTIONAL ANS [all], MAXLEN=50 **Validated:** Yes **Example:** John Smith
CardNumber	**Description:** Credit card number **Format:** String **Constraints:** MANDATORY N, MINLEN=13, MAXLEN=16 **Validated:** Yes **Example:** 5123456789012346
Cvn	**Not required and will be ignored if included in the request. CVN fields are never stored in BPOINT system.**
ExpiryDateMonth	**Description:** Credit card expiry date month In MM format

	Format: **Constraints:** **Validated:** **Example:**	String MANDATORY N, MINLEN=2, MAXLEN=2 Yes 05
ExpiryDateYear	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Credit card expiry date year In YY format String MANDATORY N, MINLEN=2, MAXLEN=2 Yes 17
AccountName	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Bank account name String MANDATORY ANS [EBCDIC character set], MINLEN=1, MAXLEN=32 Yes John Smith
AccountNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Bank account number String MANDATORY AN, MINLEN=3, MAXLEN=9 Yes 00123
BSBNumber	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	BSB number String MANDATORY N, MINLEN=6, MAXLEN=6 Yes 061200
Crn1	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 1 Ignored if provided in “Create AuthKey” request String MANDATORY (OPTIONAL if provided in “Create AuthKey” request) ANS [], MAXLEN=50 Yes Customer 1234
Crn2	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 2 Ignored if provided in “Create AuthKey” request String OPTIONAL ANS [], MAXLEN=50 Yes Invoice 987
Crn3	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Customer reference number 3 Ignored if provided in “Create AuthKey” request String OPTIONAL ANS [], MAXLEN=50 Yes customer@email.com
EmailAddress	**Description:** **Format:** **Constraints:** **Validated:** **Example:**	Email address Ignored if provided in “Create AuthKey” request String OPTIONAL ANS [], MAXLEN=250 Yes customer@email.com

Response Headers:

- Location: URL to redirect to

Response JSON Object:

- None

Status Codes:

- 302 Found – Request completed successfully, redirect to URL found in response “Location” header

Request:

```
POST https://www.bpoint.com.au/webapi/v2/dvtokens/webform/update HTTP/1.1
Content-Type: application/x-www-form-urlencoded charset=utf-8
Host: www.bpoint.com.au
Content-Length: 215
Expect: 100-continue

&Crn1=test+crn1&Crn2=test+crn2&Crn3=test+crn3&EmailAddress=john%40john.com&CardNumber=512345678901234
6&ExpiryDateMonth=05&ExpiryDateYear=17&Cvn=&CardHolderName=John+Smith&AuthKey=8eeacd3c-68b0-42c4-
8566-da12e83c5d74
```

Response:

```
HTTP/1.1 302 Found
Cache-Control: private
Location: http://merchant.com/tokenreceipt?ResponseCode=0&ResponseText=Success&ResultKey=f91b29bd-d936-
432a-8f20-8b42aad5aaf7
Server: Microsoft-IIS/7.5
Set-Cookie: ASP.NET_SessionId=riadmkq0refenuwhcvjqzfvb; path=/; HttpOnly
p3p: CP="IDC DSP COR ADM DEVi TAIi PSA PSD IVAi IVDi CONi HIS OUR IND CNT"
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 21:53:02 GMT
Content-Length: 0
```

4.12 Update a DataVault Token using an AuthKey (JavaScript)

BPOINT API JavaScript makes it easy to update a DataVault token without having the information pass through your servers.

Including api.js

```
<script src="https://www.bpoint.com.au/api/cba/api.js?v=2" type="text/javascript"></script>
```

Add this script tag to your page (preferably at the bottom) to get started with the BPOINT JavaScript API. For UAT integration testing please use the URL below for the src attribute of script tag - https://bpoint-uat.premier.com.au/api/cba/api.js?v=2

Setting up add DataVault token form

```
CBA.SetupUpdateToken({
            AppendToElementId: "update-form-location",
            AuthKey: $("#AuthKey").val(),
            DefaultErrorUrl: "https://www.yourdomain.com/handleerror"
});
```

This is all you need to get started with updating a DataVault token. The BPOINT JavaScript library will update the card details and will redirect the browser to the "RedirectionUrl" supplied when the AuthKey was created (Refer 4.8). At the end of the redirection, invoke the "Lookup Add/Update DataVault Token Result" call (Refer 4.13) from your web server to retrieve the DataVault token information and render the confirmation receipt to your customers.

NOTE: Whilst this example uses jQuery's val() to retrieve values, you can also use standard DOM methods to retrieve card data from your form. The BPOINT JavaScript API is JavaScript library agnostic.

The three parameters in the sample code above are compulsory. There are a number of optional parameters which you can pass to the "SetupUpdateToken" method to further customise the form as required. Please find below information about the parameters which you can pass to "SetupUpdateToken" method.

Name:	Comments:	
AppendToElementId	**Description:**	ID of the HTML element on your page where you want the form to be inserted.
	Format:	String
	Constraints:	MANDATORY
	Example:	"update-form-location"
AuthKey	**Description:**	Unique key created using "Create AuthKey" operation (Refer 4.8). You may store this in a hidden field and then pass the value.
	Format:	String
	Constraints:	MANDATORY
	Example:	$("#AuthKey").val()
DefaultErrorUrl	**Description:**	A fall-back URL which the browser is redirected to if a response is not received.
	Format:	String
	Constraints:	MANDATORY
	Example:	"https://www.yourdomain.com/handleerror"
Crn1	**Description:**	Include this object with appropriate flags if you wish to display customer reference number 1 field on the page ǂ
	Format:	Object
	Constraints:	OPTIONAL if supplied while creating the AuthKey (Refer 4.8)
	Example:	{Visible: true, LabelName: "Reference1", Value: "", ReadOnly: false}
Crn2	**Description:**	Include this object with appropriate flags if you wish to display customer reference number 2 field on the page ǂ

	Format:	Object
	Constraints:	OPTIONAL
	Example:	{Visible: true, LabelName: “Reference2”, Value: “”, ReadOnly: false}
Crn3	**Description:**	Include this object with appropriate flags if you wish to display customer reference number 3 field on the page ǂ
	Format:	Object
	Constraints:	OPTIONAL
	Example:	{Visible: true, LabelName: “Reference3”, Value: “”, ReadOnly: false}
EmailAddress	**Description:**	Include this object with appropriate flags if you wish to display email address field on the page ǂ
	Format:	Object
	Constraints:	OPTIONAL
	Example:	{Visible: true, LabelName: “Email”, Value: “”, ReadOnly: false}
Type	**Description:**	The type of information that you wish to store / change. E.g. Credit card or bank account. You may also choose to offer this choice to your customer.
	Format:	String
	Constraints:	OPTIONAL
	Example:	CREDITCARD – this will render only the credit card fields on the form BANKACCOUNT – this will render only the bank account fields on the form OPTION – This will render the form with radio buttons, offering the user the option to choose which type of information they want to store.
AcceptBankAccountTerms	**Description:**	This has to be true for the Bank account token type. This notifies BPOINT server that you have rendered appropriate terms and conditions to the user and the user has accepted. This flag can be ignored for the credit card token type.
	Format:	Boolean
	Constraints:	MANDATORY if storing Bank account tokens, optional otherwise.
	Example:	true

ǂ if this parameter is populated via “Create Auth Key” (Refer 4.8) operation then that value will take precedence over value submitted here.

“SetupUpdateToken” as explained above is the simple approach to start updating previously stored credit card or bank account information using BPOINT API. For more control over the user experience, please follow the custom method as described below.

Process update DataVault token

Invoke the process update token method as explained below when you are ready to update the DataVault token e.g. a button click. “ProcessUpdateToken” is an asynchronous call – it returns immediately and invokes the “CallbackFunction” when it received a response from the BPOINT server.

```
CBA.ProcessUpdateToken({
        AuthKey: $("#AuthKey").val(),
        Crn1: $("#CRN1").val(),
        Crn2: $("#CRN2").val(),
        Crn3: $("#CRN3").val(),
        EmailAddress: $("#EmailAddress").val(),
        Type: “CREDITCARD”, //Either CREDITCARD or BANKACCOUNT
        CardHolderName: $("#CardHolderName").val(),
        CardNumber: $("#CardNumber").val(),
        ExpiryMonth: $("#CardExpiryMonth").val(),
        ExpiryYear: $("#CardExpiryYear").val(),
        BSBNumber: $("#BsbNumber").val(),
        BankAccountNumber: $("#BankAccountNumber").val(),
        BankAccountName: $("#BankAccountName").val(),
        AcceptBankAccountTerms: $("#AcceptBankAccountTerms").prop("checked"),
        CallbackFunction: ProcessUpdateTokenCallBack
});
```

Please find below information on the parameters which you can pass in to “ProcessUpdateToken” method.

Name:	Comments:	
AuthKey	**Description:**	Unique key created using “Create AuthKey” operation (Refer 4.8). You may store this in a hidden field and then pass the value.
	Format:	String
	Constraints:	MANDATORY
Crn1	**Description:**	Customer reference number 1 to update against the DataVault token. Pass null if you do not wish to update this field.
	Format:	String
	Constraints:	OPTIONAL if supplied while creating the AuthKey (Refer 4.8)
Crn2	**Description:**	Customer reference number 2 to update with the DataVault token. Pass null if you do not wish to update this field.
	Format:	String
	Constraints:	OPTIONAL
Crn3	**Description:**	Customer reference number 3 to update with the DataVault token. Pass null if you do not wish to update this field.
	Format:	String
	Constraints:	OPTIONAL
EmailAddress	**Description:**	Email address to update with the DataVault token. Pass null if you do not wish to update this field.
	Format:	String
	Constraints:	OPTIONAL
Type	**Description:**	The type of information that you wish to store / change. E.g. Credit

		card or bank account. You may also choose to offer this choice to your customer via radio buttons.
	Format:	String
	Constraints:	OPTIONAL (Default is CREDITCARD)
	Value	Either CREDITCARD or BANKACCOUNT
CardHolderName	**Description:**	Name on the card
	Format:	String
	Constraints:	OPTIONAL
CardNumber	**Description:**	Card number you wish to store
	Format:	Numeric
	Constraints:	MANDATORY if TYPE is CREDITCARD
ExpiryMonth	**Description:**	Expiry month of the card
	Format:	Numeric
	Constraints:	MANDATORY if TYPE is CREDITCARD
ExpiryYear	**Description:**	Expiry year of the card
	Format:	Numeric
	Constraints:	MANDATORY if TYPE is CREDITCARD
BSBNumber	**Description:**	BSB number you wish to store
	Format:	String
	Constraints:	MANDATORY if TYPE is BANKACCOUNT
BankAccountNumber	**Description:**	Bank account number you wish to store
	Format:	String
	Constraints:	MANDATORY if TYPE is BANKACCOUNT
BankAccountName	**Description:**	Bank account you wish to store
	Format:	String
	Constraints:	MANDATORY if TYPE is BANKACCOUNT
AcceptBankAccountTerms	**Description:**	This must be true for the Bank account token type. This notifies BPOINT server that you have rendered appropriate terms and conditions to the user and the user has accepted. This flag can be ignored for the credit card token type.
	Format:	Boolean (true or false)
	Constraints:	MANDATORY if TYPE is BANKACCOUNT
CallbackFunction	**Description:**	This is a callback you provide to handle the response from the BPOINT API. Please see below for more information on this.
	Format:	Function
	Constraints:	MANDATORY

The “CallbackFunction” is a JavaScript function you provide to handle the response from the BPOINT API. It does the following

1. If the payment information entered by user returned an error, display it on the page

2. If an error was not returned, then submit the result back to your server to call “Lookup Add/Update DataVault Token Result” (Refer 4.13) and render the conformation receipt. **Note:** It is recommended that you do not submit the card details to your server, and only submit the result of the call back to your server.

Below is a sample implementation of “ProcessUpdateTokenCallBack”:

```
function ProcessUpdateTokenCallBack(result) {
        var errors = new Array();

        if (result.AjaxResponseType == 0) { //AJAX call was successful

                if (result.ApiResponseCode == 0) {
                //API returned success. Refer to Appendix 6.1 for API Response codes.
                //Submit result.ResultKey to your server for further processing(Refer 4.13)
                }
                else {
                        errors = result.Errors;
                }
        }
        else if(result.AjaxResponseType == 1) { //Error with AJAX call
                errors = result.Errors;
        }
        else if(result.AjaxResponseType == 2) { //AJAX call timed out
                errors = result.Errors;
        }

        //Show errors on the page
        if (errors.length > 0) {
                var ul = $("<ul></ul>");

                $.each(errors, function (i, r) {
                        ul.append("<li>" + r.Message + "</li>");
                });

                $(".validation-summary").append(ul).show();
        }
}
```

The result object is described as below:

Name:	**Comments:**
AjaxResponseType	**Description:** This is the result of the AJAX call. This will assist you in handling the call back. **Format:** Numeric **Values:** Possible values are 0, 1 or 2. Where 0 is success, 1 is error and 2 is timeout.
ApiResponseCode	**Description:** This is the response code returned by the BPOINT API **Format:** Numeric **Values:** Please refer to Appendix 6.1 for possible values
Errors	**Description:** List of errors returned, e.g. validation errors. **Format:** Array of object {Message: “Invalid card number”,PropertyName: “CardNumber”}

ResultKey	**Description:** This is the result key that you will have to submit to retrieve the DataVault token information, “Lookup Add/Update DataVault Token Result” (Refer 4.13). **Format:** String

4.13 Lookup Add/Update Data Vault Token Result

Retrieves result of a DataVault token added/updated via operation using Browser integration method

GET /dvtokens/withauthkey/{resultKey}

Integration:

- Merchant

URL Parameters:

- resultKey – result key returned in a response to add/update transaction operation

Request Headers:

- Authorisation: base64 encoded API user credentials

Response Headers:

- Content-Type: application/json; charset=utf-8

Response JSON Object:

Name:	Comments:
APIResponse	See section 2.3.1
DVTokenResp	See section 2.4.3.6

Status Codes:

- 200 OK – Request completed successfully

Request:

```
GET https://www.bpoint.com.au/webapi/v2/dvtokens/withauthkey/208e4a07-ac75-479e-ac4e-66ada2066f00 HTTP/1.1
Authorization: YXBpdGVzdHwwMDAwMDAwMDAwMDAwMDAwOnBhc3N3b3Jk
Host: www.bpoint.com.au
```

Response:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 320
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 06 Nov 2014 21:42:47 GMT
```

```
{
	"APIResponse" : {
		"ResponseCode" : 0,
		"ResponseText" : "Success"
	},
	"DVTokenResp" : {
		"BankAccountDetails" : null,
		"CardDetails" : {
			"CardHolderName" : "John Smith",
			"ExpiryDate" : "0517",
			"MaskedCardNumber" : "512345...346"
		},
		"CardType" : "MC",
		"Crn1" : "12345",
		"Crn2" : "",
		"Crn3" : "",
		"EmailAddress" : "cardholder@email.com.au",
		"DVToken" : "5999991550629267"
	}
}
```

5 WebHooks

WebHooks are an asynchronous notification mechanism. They are triggered by operations invoked using the Browser integration method. They notify merchant systems that a successful operation has taken place and provides the detailed result.

WebHooks are triggered only if the merchant has provided a WebHook URL in the request to obtain an AuthKey. The BPOINT system will attempt the WebHook notification until successfully acknowledged by the merchant or the retry period has expired.

To acknowledge a WebHook notification, the merchant's application must respond to the request with the HTTP Status Code 200.

Example WebHook Request Triggered by Process Transaction operation

```
POST http://merchant.com/txnWebHook HTTP/1.1
Content-Type: application/json; charset=utf-8
Host: merchant.com
Content-Length: 827
Expect: 100-continue
Proxy-Connection: Keep-Alive

{
	"Action" : "payment",
	"Amount" : 19900,
	"AmountSurcharge" : 0,
	"AmountSurchargeGST" : 0,
	"AuthoriseId" : "461246",
	"BankAccountDetails" : null,
	"BankResponseCode" : "00",
	"BillerCode" : "",
	"CVNResult" : {
		"CVNResultCode" : "Unsupported"
	},
	"CardDetails" : {
		"CardHolderName" : "John Smith",
		"ExpiryDate" : "0517",
		"MaskedCardNumber" : "512345...346"
	},
	"CardType" : "MC",
	"Crn1" : "test crn1",
	"Crn2" : "test crn2",
	"Crn3" : "test crn3",
	"Currency" : null,
	"IsCVNPresent" : true,
	"IsThreeDS" : false,
	"MerchantNumber" : "0000000000000000",
	"MerchantReference" : "test merchant ref",
	"OriginalTxnNumber" : null,
	"ProcessedDateTime" : "2014-11-06T12:59:50.2200000",
	"RRN" : "431109461246",
	"ReceiptNumber" : "48368876603",
	"ResponseCode" : "0",
	"ResponseText" : "Approved",
	"SettlementDate" : "20141107",
	"Source" : "api",
	"StoreCard" : false,
	"SubType" : "single",
	"ThreeDSResponse" : null,
	"TxnNumber" : "46476603",
	"Type" : "internet"
}
```

6 Appendix

6.1 API Response Codes

Response Code:	Description:
0	Success
1	Invalid credentials
2	Invalid permissions
3	User not found
101	Invalid field: action
102	Invalid field: type
103	Invalid field: subtype
104	Invalid field: merchant number
105	Invalid field: biller code
106	Invalid field: CRN1
107	Invalid field: CRN2
108	Invalid field: CRN3
109	Invalid field: currency
110	Invalid field: amount
111	Invalid field: merchant reference
112	Invalid field: card number
113	Invalid field: card holder name
114	Invalid field: expiry date
115	Invalid field: CVN
116	Invalid field: web hook URL
117	Invalid field: redirection URL
118	Invalid field: transaction number
119	Invalid field: original transaction number
120	Invalid field: receipt number
121	Invalid field: settlement date
122	Invalid field: masked card number
123	Invalid field: DVToken
124	Invalid field: bank account number
125	Invalid field: BSB number
126	Invalid field: bank account name
127	Invalid field: email address
128	Invalid field: store card
201	Transaction not found
202	DVToken not found
203	Transaction type cannot be tokenised
204	Transactions cannot be tokenised because card holder has not given permission
205	Biller code not found
206	Session not found
207	Invalid session
208	Transaction must be approved to be tokenised
209	Search returned no results
210	Merchant details not found
211	Merchant account settings not found
300	Follow redirection

999	Fatal error

6.2 Transaction Actions

Action:	Description:
payment	A financial transaction that debits the funds from the card.
refund	A financial transaction that credits the funds to the card. This applied only to “matched” refunds where the funds are returned to the card used in the original payment transaction.
unmatched_refund	A financial transaction that credits the funds to the nominated card. This refund is not “matched” to the card used in the original payment transaction. Unmatched refunds are not available as a standard service. Please contact your bank representative if you require this service.
preauth	An authorisation transaction that reserves the funds against the card but does not debit the card. The cardholder’s available balance will decrease by the preauth amount if the preauth is approved. A preauth typically remains against the card for 4-7 days depending on the Issuer. Expiry of the preauth expires the funds reservation.
capture	A financial transaction that is linked to a preauth. This debits the card immediately.
reversal	Reverses the original transaction so that both the original transaction and the reversal will not appear on the card statement. **Note:** A reversal can only be processed before the end of day cut-off on the same day as the original transaction.
de_rejection	A bank account direct debit that has been returned by the account holder’s bank. The funds will be debited from your trace account.

6.3 Transaction Types

Type:	Description:
callcentre	Used for a call centre transaction. Transaction is flagged as a telephone order to the Issuer.
cardpresent	Used for reporting purposes only. Transaction is flagged as a telephone order to the Issuer.
ecommerce	Used for a real time customer internet transaction. Transaction is flagged as an internet order to the Issuer.
internet	Used for a real time customer internet transaction. Transaction is flagged as an internet order to the Issuer.
ivr	Used for an automated phone (IVR) transaction. Transaction is flagged as a telephone order to the Issuer.
mailorder	Used for a mail order transaction. Transaction is flagged as a mail order to the Issuer.
telephoneorder	Used for a telephone order transaction. Transactions is flagged as a telephone order to the Issuer.

6.4 Transaction Sub Types

Sub Type:	Description:
single	Used for one-off transactions
recurring	Used when the merchant regularly debits the account, e.g. monthly billing.

6.5 Card Types

Card Type:	Description:
AE	American Express
DC	Diners Club
JC	JCB Card
MC	MasterCard
VC	Visa
BA	Bank Account

6.6 Sources

Source:	Description:
api	Transaction submitted using the API
callcentre	Transaction submitted by the call centre
customerportal	Transaction submitted through the Consumer Portal
internet	Transaction submitted through the hosted payment page
ishop	Transaction submitted through the iSHOP
ivr	Transaction submitted through hosted IVR
backoffice	Transaction submitted through the Merchant Back Office
mobilebackoffice	Transaction submitted through the Mobile Back Office
sftp	Transaction submitted via SFTP

6.7 CVN Responses

Code:	Description:
M	Valid or matched CVN
S	Merchant indicates CVN not present on card
P	CVN Not Processed
U	Card issuer is not registered and/or certified
N	Code invalid or not matched
Unsupported	Acquiring institution does not support CVN

6.8 ThreeDS ECI Responses

Code:	Description:
05	Cardholder authenticated
06	Cardholder not enrolled
Note: The values may change depending on the locale or Issuer.	

6.9 ThreeDS Enrolled Responses

Code:	Description:
Y	Yes, cardholder is enrolled
N	No, cardholder is not enrolled
U	Unavailable to check

6.10 ThreeDS Status Responses

Code:	Description:
Y	Yes, cardholder authenticated
N	No, cardholder was not authenticated
A	Attempted Authentication
U	Unavailable to check

6.11 ThreeDS Verify Security Level Responses

Code:	Description:
0	MasterCard – merchant is not participating in ThreeDS
1	MasterCard – Cardholder is not participating in ThreeDS
2	MasterCard – Cardholder authenticated
05	Visa – Fully authenticated Amex – Fully authenticated
06	Visa – Not authenticated (cardholder is not participating in ThreeDS) Amex – Not authenticated (cardholder is enrolled, but authentication failed)
07	Visa – Note authenticated. Usally related to a system issue. Amex – Not authenticated.

6.12 ThreeDS Verify Status Responses

Code:	Description:
Y	Cardholder successfully authenticated
M	Cardholder is not enrolled, but the Issuer attempted processing
E	Cardholder is not enrolled
F	Request format error
N	Verification failed
S	The signature on the response received from the Issuer could not be validated
P	Error receiving input from Issuer
I	Internal error
U	Verification was unable to be completed. This can be caused by network or system failures
T	The cardholder session timed out and the browser did not return from the Issuer's ThreeDS site
A	Merchant authentication failed
D	Communication error
C	Card type not supported

6.13 ThreeDS Verify Type Responses

Code:	Description:
3DS	ThreeDS
SPA	Secure Payment Authentication from MasterCard

6.14 TxnResp Responses

Response Code:	Bank Response Code:	Response Text:
Bank Response Codes:		
0	00	Approved
0	08	Honour with ID
0	16	Approved, Update Track 3
1	09	Transaction Declined - Bank Error
1	10	Transaction Declined - Bank Error
1	11	Transaction Declined - Bank Error
1	12	Transaction Declined - Bank Error
1	13	Transaction Declined - Bank Error
1	17	Transaction Declined - Bank Error
1	18	Transaction Declined - Bank Error
1	20	Transaction Declined - Bank Error
1	21	Transaction Declined - Bank Error
1	22	Transaction Declined - Bank Error
1	23	Transaction Declined - Bank Error
1	24	Transaction Declined - Bank Error
1	26	Transaction Declined - Bank Error
1	27	Transaction Declined - Bank Error
1	28	Transaction Declined - Bank Error
1	29	Transaction Declined - Bank Error
1	30	Transaction Declined - Bank Error
1	32	Transaction Declined - Bank Error
1	35	Transaction Declined - Bank Error
1	37	Transaction Declined - Bank Error
1	38	Transaction Declined - Bank Error
1	40	Transaction Declined - Bank Error
1	42	Transaction Declined - Bank Error
1	44	Transaction Declined - Bank Error
1	45	Transaction Declined - Bank Error
1	46	Transaction Declined - Bank Error
1	47	Transaction Declined - Bank Error
1	48	Transaction Declined - Bank Error
1	49	Transaction Declined - Bank Error
1	50	Transaction Declined - Bank Error
1	52	Transaction Declined - Bank Error
1	53	Transaction Declined - Bank Error
1	55	Transaction Declined - Bank Error
1	56	Transaction Declined - Bank Error
1	57	Transaction Declined - Bank Error
1	58	Transaction Declined - Bank Error
1	60	Transaction Declined - Bank Error
1	62	Transaction Declined - Bank Error
1	63	Transaction Declined - Bank Error
1	64	Transaction Declined - Bank Error
1	66	Transaction Declined - Bank Error
1	67	Transaction Declined - Bank Error
1	69	Transaction Declined - Bank Error
1	70	Transaction Declined - Bank Error
1	71	Transaction Declined - Bank Error
1	72	Transaction Declined - Bank Error
1	73	Transaction Declined - Bank Error
1	74	Transaction Declined - Bank Error
1	75	Transaction Declined - Bank Error
1	76	Transaction Declined - Bank Error
1	77	Transaction Declined - Bank Error
1	78	Transaction Declined - Bank Error
1	79	Transaction Declined - Bank Error
1	80	Transaction Declined - Bank Error

1	**81**	Transaction Declined - Bank Error
1	**82**	Transaction Declined - Bank Error
1	**83**	Transaction Declined - Bank Error
1	**84**	Transaction Declined - Bank Error
1	**85**	Transaction Declined - Bank Error
1	**86**	Transaction Declined - Bank Error
1	**87**	Transaction Declined - Bank Error
1	**88**	Transaction Declined - Bank Error
1	**89**	Transaction Declined - Bank Error
1	**93**	Transaction Declined - Bank Error
1	**94**	Transaction Declined - Bank Error
1	**95**	Transaction Declined - Bank Error
1	**96**	Transaction Declined - Bank Error
1	**97**	Transaction Declined - Bank Error
2	**01**	Bank Declined Transaction
2	**02**	Bank Declined Transaction
2	**03**	Bank Declined Transaction
2	**04**	Bank Declined Transaction
2	**05**	Bank Declined Transaction
2	**06**	Bank Declined Transaction
2	**07**	Bank Declined Transaction
2	**14**	Bank Declined Transaction
2	**15**	Bank Declined Transaction
2	**19**	Bank Declined Transaction
2	**25**	Bank Declined Transaction
2	**31**	Bank Declined Transaction
2	**34**	Bank Declined Transaction
2	**36**	Bank Declined Transaction
2	**39**	Bank Declined Transaction
2	**41**	Bank Declined Transaction
2	**43**	Bank Declined Transaction
2	**59**	Bank Declined Transaction
2	**61**	Bank Declined Transaction
2	**65**	Bank Declined Transaction
2	**90**	Bank Declined Transaction
2	**91**	Bank Declined Transaction
2	**92**	Bank Declined Transaction
2	**98**	Bank Declined Transaction
2	**99**	Bank Declined Transaction
3	**68**	Transaction Declined - No Reply from Bank
4	**33**	Transaction Declined – Expired Card
4	**54**	Transaction Declined – Expired Card
5	**51**	Bank Declined Transaction
Gateway response codes:		
?		Response Unknown
6		Transaction Declined - Error Communicating with Bank
7		Payment Server Processing Error - Typically caused by invalid input data such as an invalid credit card number. Processing errors can also occur
8		Transaction Declined - Transaction Type Not Supported
9		Bank Declined Transaction (Do not contact Bank)
A		Transaction Aborted
C		Transaction Cancelled
D		Deferred Transaction
E		Issuer Returned a Referral Response
F		3D Secure Authentication Failed
I		Card Security Code Failed
L		Shopping Transaction Locked (This indicates that there is another transaction taking place using the same shopping transaction number)
N		Cardholder is not enrolled in 3D Secure (Authentication Only)
P		Transaction is Pending
R		Retry Limits Exceeded, Transaction Not Processed
S		Duplicate OrderInfo used. (This is only relevant for Payment Servers that enforce the

		uniqueness of this field)
U		Card Security Code Failed
Payment Server Response Codes		
?		Unhandled server error.
PT_E1		Database error.
PT_E2		Unable to encrypt card number.
PT_E3		Unable to decrypt card number.
PT_E4		Server shutdown in progress.
PT_E5		Server busy, transaction timed out in queue and was not sent to the bank.
PT_E6		Processing aborted, payment server is shutting down.
PT_V1		Invalid transaction type.
PT_V2		Invalid financial type.
PT_V3		Invalid amount.
PT_V4		Invalid card number.
PT_V5		Invalid expiry date.
PT_V6		Invalid CVN.
PT_V7		Financial transaction type not supported by gateway.
PT_V8		Reversal not supported.
PT_V9		Merchant/biller details not found.
PT_V10		Unable to retrieve merchant/biller details.
PT_V11		Cardholder not authenticated (Vbv, SecureCode).
PT_V12		Error authenticating cardholder (Vbv, SecureCode).
PT_T1		Token payment not allowed for Internet, IVR and call centre transaction types.
PT_T2		Credit Card payment details not found for this token.
PT_T3		Unable to decrypt card number.
PT_T4		Unable to retrieve credit card payment details due to system error.
PT_T5		Token payment not supported.
PT_R1		Original transaction not found.
PT_R2		Original transaction was not approved.
PT_R3		Original transaction is locked.
PT_R4		Transaction already fully refunded.
PT_R5		Only $x.xx available for refund.
PT_R6		Preauth transaction already completed.
PT_R7		Unable to verify if reversal can be processed.
PT_R8		Transaction already reversed.
PT_R9		Transaction partially refunded.
PT_R10		(Only for reversals of timed out transactions) Original transaction not found.
PT_R11		(Only for reversals of timed out transactions) Multiple instances of original transaction found.
PT_R12		(Only for reversals of timed out transactions) Original transaction was not successful.
PT_R13		(Only for reversals of timed out transactions) Original transaction number not found.
PT_R14		(Only for reversals of timed out transactions) Error looking up result of original transaction.
PT_R15		Invalid amount. Reversal amount must be the same as the amount of the original transaction.
PT_G1		Gateway configuration error.
PT_G2		Unable to build gateway request.
PT_G3		Unable to connect to gateway.
PT_G4		Unable to send transaction request data.
PT_G5		Unable to get response data.
PT_G6		Unable to process transaction.
PT_G7		Unable to process, server busy.
PT_G8		Unable to parse response data.