



# **BPOINT Web Service API Documentation**

*Version 1.4.1*

## Contents

<b>1</b>	<b>Getting Started</b> .....	<b>3</b>
1.1	Using the webservice.....	3
1.2	Webservice URL.....	3
<b>2</b>	<b>API Web Methods</b> .....	<b>4</b>
2.1	“ProcessPayment” web method .....	4
2.2	“SearchTransactions” web method.....	6
2.3	“AddToken” web method .....	11
2.4	“GetRecentlyModifiedTokens” web method .....	12
2.5	“DeleteToken” web method .....	14
2.6	“GetUnusedTokens” web method.....	15
2.7	“UpdateToken” web method .....	16
2.8	“UpdateTokenMulti” web method .....	17
2.9	“SearchTokens” web method .....	18
2.10	“CreateSession” web method .....	20
2.11	“IsBankAvailable” web method .....	21
2.12	“UploadCRNLookupFile” web method.....	22
2.13	“UploadCRNLookupZipFile” web method.....	23
2.14	“SubmitBatch” web method .....	24
2.15	“RetrieveBatchStatistics” web method.....	25
2.16	“DownloadBatchFileByFilename” web method .....	26
2.17	“DownloadBatchFileById” web method.....	27
2.18	“RetrieveSettlementFiles” web method .....	28
2.19	“Login” web method .....	29
2.20	“Logout” web method.....	29
2.21	“GetAnonymousToken” web method.....	30
2.22	“TokenisePayment” web method.....	31
<b>3</b>	<b>Code Samples</b> .....	<b>33</b>
3.1	C-Sharp (C#) .....	33
3.2	Visual Basic (VB.NET).....	35
3.3	PHP.....	37
3.4	Java .....	39
<b>4</b>	<b>Appendix</b> .....	<b>43</b>
4.1	Transaction Response Codes .....	43

---

## 1 Getting Started

---

### 1.1 Using the webservice

The BPOINT API public webservice allows you to access all the merchant functionalities from within your own website or computer application.

To access the API, you must:

- **Have API access enabled in your merchant facility**
- **Have an API user login and password** (you may create an API user via the user management section in the back office website)

Please contact your account manager or BPOINT support for any enquires on enabling or accessing the API functionality.

For more information on how to use the API please refer to the Web Method (chapter 2) and code samples (chapter 3) sections of this document.

### 1.2 Webservice URL

The webservice URL you must call is:

**<https://www.bpoint.com.au/evolve/service.asmx>**

## 2 API Web Methods

\*\* Compulsory parameter – cannot be empty or null

### 2.1 “ProcessPayment” web method

The process payment API method is designed to give merchants access to BPOINT payment gateway in order to automatically submit payments.

**Input parameters:**

Authentication Parameters		
Parameter name	Description	Comments
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
Input Parameters		
Parameter name	Description	Comments
PaymentType**	“PAYMENT” for payment processing and “REFUND” for refund processing. Other options are not supported	This is a compulsory parameter
TxnType**	It should be set to appropriate TxnType depending on how you received the payment	This is compulsory. Passing in “NOT_SPECIFIED” will result in error
MerchantReference	It is a reference supplied by you which helps you to identify the payment	This is optional but recommended that you populate it
CRN1	It is an additional reference	This is optional
CRN2	Additional reference	This is optional
CRN3	Additional reference	This is optional
Amount	The amount you wish to process	This is in cents
CardNumber**	The credit card number you wish to charge Note: This can also be a DataVault token number.	Not required for refund
ExpiryDate**	The expiry date on card	Not required for refund
CVC	CVC / CVV of the credit card	This is optional
OriginalTransactionNumber**	This is the payment “TransactionNumber” that you get back after a successful payment	This is compulsory if “REFUND” is selected as “PaymentType” else optional

**Output parameters:**

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for "SUCCESS" before proceeding with any other checks	If "ERROR" is returned then "ResponseMessage" will contain an appropriate error message.
<b>Output Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	This is a summary response code. 0 means approved. Anything else means that the payment declined.	In case of declined you may want to check AcquirerResponseCode field for extra information on why the payment declined
AcquirerResponseCode	This is the response code issued by the payment Acquirer	
TransactionNumber	This is an unique transaction number issued per payment	You will need to use this in case you want to process an refund against this payment
ReceiptNumber	This is an unique receipt number issued per transaction	You may provide this to your customers
Authoriseld	A code issued by the acquiring bank to approve or deny the transaction. This may not always be supplied by all acquirers.	
SettlementDate	The date of payment settlement	
MaskedCardNumber	The first 6 and last 3 digits of the credit card number charged returned back to you	
CardType	The card type used to process the payment	

## 2.2 “SearchTransactions” web method

The search transaction API method is designed to give merchants access to the BPOINT payment transaction gateway in order to make it easy to integrate merchant financial systems with BPOINT.

Any search query submitted using this method will return maximum 500 results.

### Input parameters:

<b>Authentication Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
<b>Input Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
MerchantNumber	Merchant facility number of any of the child merchants	This is an optional parameter.  The API allows parent merchants to search for their child merchant transactions. This should be populated with respective child merchant number if you have any. Otherwise the search will be performed on the merchant number supplied in the Authentication Parameters.
PaymentType	Any value from the following list - NOT_SPECIFIED - PAYMENT - REFUND - CAPTURE - UNMATCHED_REFUND - PREAUTH - REVERSAL	This is an optional parameter  To search all payment types select: NOT_SPECIFIED
TxnType	Any value from the following list - NOT_SPECIFIED - MAIL_ORDER - TELEPHONE_ORDER - RECURRING - ECOMMERCE - CUSTOMER_PRESENT - CALL_CENTRE - IVR - INTERNET_ANONYMOUS	This is an optional parameter  To search all transaction types select: NOT_SPECIFIED

	- WEB_SHOP	
NumOfHours**	Number of hours in past from now to search the transactions for	<p>If populated then it should be populated with an integer value.</p> <p>E.g. If 0 is supplied then this parameter will be ignored and the ToDate and FromDate will be used instead.</p> <p>E.g. If 1 is supplied then the search will return transactions made within last 1 hour from now.</p>
FromDate**	Transaction Process date time will be compared against this parameter and all the transactions made on or after the value supplied will be returned.	<p>Only used if NumOfHours is 0</p> <p>If populated then it should be populated with Date Time value in format: "YYYY-MM-DDTHH:MM:SS"</p>
ToDate**	Transaction Process date time will be compared against this parameter and all the transactions made before the value supplied will be returned.	<p>Only used if NumOfHours is 0</p> <p>If populated then it should be populated with Date Time value in format: "YYYY-MM-DDTHH:MM:SS"</p>
ReceiptNumber	The receipt number to search for	<p>This is an optional parameter.</p> <p>If you need to look for a single transaction and have a BPOINT receipt number for the transaction you can supply it here.</p>
MerchantReference	It is a reference supplied when a payment was made	This is an optional parameter
CRN1	It is an additional reference supplied when a payment was made	This is an optional parameter
CRN2	It is an additional reference supplied when a payment was made	This is an optional parameter
CRN3	It is an additional reference supplied when a payment was made	This is an optional parameter
Amount	The amount of the transaction	<p>This is an optional parameter.</p> <p>If supplied this should be a cents value.</p> <p>E.g. You need to supply 100 if you are looking for transactions worth \$1.00</p> <p>Leave as '0' to search all amounts.</p>
MaskedCardNumber	The credit card number used to	This is an optional

	process the transaction	parameter.  If supplied then it should be populated with either first 6 digits of the credit card or credit card number in a masked form E.g. 444433...111
ExpiryDate	The expiry date of the credit card used to process the transaction	This is an optional parameter.  If supplied then it should be populated with card expiry date in format MMY
ResponseCode	The response code to search	This is an optional parameter.  This is a summary response code. 0 means approved. Anything else means that the payment declined.  E.g. 0 will search for all "APPROVED" transaction
AcquirerResponseCode	The Acquirer response code to search for	This is an optional parameter.  This is the response code issued by the payment Acquirer.  E.g. 00, 08 are approved.
TransactionNumber	The transaction to search for	This is an optional parameter.  This is a unique transaction number issued per payment
Authoriseld	Payment AuthorisedId to search for	This is an optional parameter.  This is a code issued by the acquiring bank to approve or deny the transaction. This may not always be supplied by all acquirers.
SettlementDate	Search transactions for the supplied settlement date	This is an optional parameter.  If supplied then it should be form "YYYY-MM-DDTHH:MM:SS"
CardType	The credit card type to search for	This is an optional parameter.  It can be populated with any



		of the following values: - MC - VC - AE - JC - DC
BillerCode	The biller code to search the transactions for	This is an optional parameter.  This is the BPOINT biller code linked to your merchant facility.
SearchOrder	The order in which the results should be returned. It can be any value from the following list: - ASCENDING - DESCENDING	This is an optional parameter.

#### Output parameters:

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for "SUCCESS" before proceeding with any other checks	If "ERROR" is returned then "ResponseMessage" will contain an appropriate error message.
<b>Output Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
PaymentType	Any value from the following list - NOT_SPECIFIED - PAYMENT - REFUND - CAPTURE - UNMATCHED_REFUND - PREAUTH - REVERSAL	
TxnType	Any value from the following list - NOT_SPECIFIED - MAIL_ORDER - TELEPHONE_ORDER - RECURRING - ECOMMERCE - CUSTOMER_PRESENT - CALL_CENTRE - IVR - INTERNET_ANONYMOUS - WEB_SHOP	
MerchantReference	It is a reference supplied when a	

	payment was made	
CRN1	It is an additional reference supplied when a payment was made	
CRN2	It is an additional reference supplied when a payment was made	
CRN3	It is an additional reference supplied when a payment was made	
Amount	The transaction amount	This is a cent value
CardNumber	The credit card number used for the transaction	This is a masked value. E.g. 444433...111
ExpiryDate	Expiry Date of the card used to process the transaction	Value returned in format MMY Y
CVC	This field is not used at this stage	
OriginalTxnNumber	Unique Transaction number issued per payment	This field will contain a value if the transaction is a REFUND transaction otherwise it will be blank.  In case of a REFUND transaction this field will contain value of the Transaction number of the Payment which was refunded.
ResponseCode	This is a summary response code. 0 means approved. Anything else means that the payment declined.	In case of declined you may want to check AcquirerResponseCode field for extra information on why the payment declined
AcquirerResponseCode	This is the response code issued by the payment Acquirer	
AuthorisationResult	This field contains description respective to the ResponseCode field	
TransactionNumber	This is an unique transaction number issued per payment	You will need to use this in case you want to process an refund against this payment
ReceiptNumber	This is an unique receipt number issued per transaction	You may provide this to your customers
Authoriseld	A code issued by the acquiring bank to approve or deny the transaction. This may not always be supplied by all acquirers.	You may ignore it
SettlementDate	The date of payment settlement	
MaskedCardNumber	The first 6 and last 3 digits of the credit card number charged returned back to you	
CardType	The card type used to process the payment	
ProcessedDate	Transaction process time	

## 2.3 “AddToken” web method

The add token API method is designed to give merchants the ability to store a customer’s card details via a token number.

### Input parameters:

Authentication Parameters		
Parameter name	Description	Comments
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
Input Parameters (TokenRequest)		
Parameter name	Description	Comments
CardNumber**	The credit card number you wish to charge	
CRN1**	It is an additional reference	
CRN2	Additional reference	This is optional
CRN3	Additional reference	This is optional
ExpiryDate**	The expiry date on card	The format is MMY

### Output parameters:

Web Service Response Parameters		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
Output Parameters (Token Response)		
Parameter name	Description	Comments
CardType	The card type used to process the payment	Eg. “MC” = mastercard “VC” = visacard
MaskedCardNumber	The first 6 and last 3 digits of the credit card number charged returned back to you	
Token	The token number of the token that was added to the data vault.	

## 2.4 “GetRecentlyModifiedTokens” web method

The getRecentlyModifiedTokens API method is designed to give merchants the ability to retrieve a list of recently modified tokens (tokens that were recently edited).

### Input parameters:

Authentication Parameters		
Parameter name	Description	Comments
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
Input Parameters		
Parameter name	Description	Comments
timePeriod**	The number of days to return. Eg. Tokens in the last week will be 7	
numOfTokens**	The maximum number of tokens to be returned	

### Output parameters:

Web Service Response Parameters		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
Output Parameters (DVTOKEN[ ])		
Parameter name	Description	Comments
CardNumber	Credit card number for the relating token.	For security reasons will always be null.
CardType	The card type used to process the payment	Eg. “MC” = mastercard “VC” = visacard
CreatedDate	Date the token was created	
CreatedUsername	Username of the person that created the token	
CRN1	It is an additional reference	
CRN2	Additional reference	
CRN3	Additional reference	
ExpiryDate	The expiry date on card	
MaskedCardNumber	The first 6 and last 3 digits of	

	the credit card number charged returned back to you	
Token	Token Number	
UpdatedDate	Date last updated	
UpdatedUsername	Username of the person that last updated the token	

## 2.5 “DeleteToken” web method

The delete token API method is designed to give merchants the ability to delete any existing tokens currently stored in the data vault.

### Input parameters:

Authentication Parameters		
Parameter name	Description	Comments
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
Input Parameters		
Parameter name	Description	Comments
token**	The token number of the token to be deleted.	The token number must exist in the data vault.

### Output parameters:

Web Service Response Parameters		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.

## 2.6 “GetUnusedTokens” web method

The getUnusedTokens API method is designed to give merchants the ability to retrieve a list of tokens that have never been used in a transaction.

### Input parameters:

Authentication Parameters		
Parameter name	Description	Comments
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
Input Parameters		
Parameter name	Description	Comments
timePeriod**	The number of days to search back.  Eg. Tokens in the last week will be 7	

### Output parameters:

Web Service Response Parameters		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
Output Parameters (DVTOKEN[ ])		
Parameter name	Description	Comments
CardNumber	Credit card number for the relating token.	For security reasons will always be null.
CardType	The card type used to process the payment	Eg. “MC” = mastercard “VC” = visacard
CreatedDate	Date the token was created	
CreatedUsername	Username of the person that created the token	
CRN1	It is an additional reference	
CRN2	Additional reference	
CRN3	Additional reference	
ExpiryDate	The expiry date on card	
MaskedCardNumber	The first 6 and last 3 digits of the credit card number charged returned back to you	
Token	Token Number	
UpdatedDate	Date last updated	
UpdatedUsername	Username of the person that last updated the token	

## 2.7 “UpdateToken” web method

The update token API method is designed to give merchants the ability to update an existing token.

### Input parameters:

Authentication Parameters		
Parameter name	Description	Comments
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
Input Parameters (TokenRequest)		
Parameter name	Description	Comments
Token**	The token number of the token to be updated	
CardNumber**	The credit card number you wish to charge	
CRN1	It is an additional reference	If this parameter is empty, then CRN1 will default to its current value.
CRN2	Additional reference	This is optional
CRN3	Additional reference	This is optional
ExpiryDate**	The expiry date on card	The format is MMY

### Output parameters:

Web Service Response Parameters		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
Output Parameters (Token Response)		
Parameter name	Description	Comments
CardType	The card type used to process the payment	Eg. “MC” = mastercard “VC” = visacard
MaskedCardNumber	The first 6 and last 3 digits of the credit card number charged returned back to you	
Token	The token number of the token that was updated.	



## 2.8 “UpdateTokenMulti” web method

The UpdateTokenMulti API method is designed to give merchants the ability to update all existing tokens with the same credit card details.

### Input parameters:

Authentication Parameters		
Parameter name	Description	Comments
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
Input Parameters		
Parameter name	Description	Comments
Token**	The token number of the token to be updated	The card number in this token will be used as the number to update all other tokens with the same number.
CardNumber**	The credit card number you wish to change to	
ExpiryDate**	The expiry date on card	The format is MMY

### Output parameters:

Web Service Response Parameters		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
Output Parameters (Token Response)		
Parameter name	Description	Comments
CardType	The card type used to process the payment	Eg. “MC” = mastercard “VC” = visacard
MaskedCardNumber	The first 6 and last 3 digits of the credit card number charged returned back to you	
Token	The token number of the token that was updated.	

## 2.9 “SearchTokens” web method

The SearchToken API method is designed to give merchants the ability to search for any token in the datavault.

### Input parameters:

<b>Authentication Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
<b>Input Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Token	The token number to be searched for.	
CRN1	The CRN to be searched for	
CRN2	The CRN to be searched for	
CRN3	The CRN to be searched for	
MaskedCardNumber	Truncated card number.	This is a masked value. E.g. “444433...111”
ExpiryDate	Expiry Date of the card	
FromDate	The minimum date to search from.	If populated then it should be populated with Date Time value in format: “YYYY-MM-DDTHH:MM:SS”
ToDate	The maximum date to search for.	If populated then it should be populated with Date Time value in format: “YYYY-MM-DDTHH:MM:SS”
CardType	The card type used to process the payment	Eg. “MC” = mastercard “VC” = visacard
ExpiredCardsOnly	Will return only tokens where the expiry date of the card, has expired.	True or False
TxnSource	The transaction type	To search all tokens, select the transaction source as “UNSPECIFIED”
UsernameCreated	Username of the user that created the token	
UsernameUpdated	Username of the user that last updated the token.	

### Output parameters:

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for “SUCCESS” before	If “ERROR” is returned then “ResponseMessage” will contain an

	proceeding with any other checks	appropriate error message.
ResponseMessage		
<b>Output Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
CardNumber		This field will be empty
ExpiryDate	Expiry date of the card	Card expiry date in MMY format
CRN1	CRN1 stored with the token	
CRN2	CRN2 stored with the token	
CRN3	CRN3 stored with the token	
Token	The token number	E.g. 5999991172546162
MaskedCardNumber	Truncated card number	This is a masked value. E.g. "444433...111"
CardType	Card Type of the stored card	Eg. "MC" = mastercard "VC" = visacard
UpdatedUsername	Username of the user that last updated the token.	
UpdatedDate	Date time when the token was updated last	
CreatedUsername	Username of the user that created the token	
createdDate	Date time when the token was created	

## 2.10 “CreateSession” web method

The CreateSession API method is designed to give merchants the ability to allow customers to redirect to the BPoint website in order to anonymously add and update tokens.

Once a SessionID has been generated it may be used to redirect to the Bpoint website.

*For more information on redirecting, refer to the ‘Anonymous Access to the BPoint Data Vault’ document.*

**Input parameters:**

<b>Authentication Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
<b>Input Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ipAddress**	IP Address of the machine where the request is coming from. If the request is initiated by browser then this field should be populated with the client (browser) IP Address.	
sharedSecret**	This is a special password shared with merchants who has access to this facility.	

**Output parameters:**

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
<b>Output Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
sessionID	The sessionID used in the query string when redirecting.	

## 2.11 “IsBankAvailable” web method

The IsBankAvailableAPI method is designed to give merchants the ability to check the availability of the bank in order to process their transactions.

### Input parameters:

Input Parameters		
Parameter name	Description	Comments
BankID**	ID of the bank	

### Output parameters:

Web Service Response Parameters		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
Output Parameters		
Parameter name	Description	Comments
Bool	True = “Available” False = “Unavailable”	

## 2.12 “UploadCRNLookupFile” web method

The uploadCRNLoopupfile API method is designed to give merchants the ability to upload a CRN lookup file onto the server. Note once a CRN lookup file has been uploaded all previous CRN's from this merchant is deleted.

### Input parameters:

<b>Authentication Parameters</b>		
There are no authentication parameters, but in order to use this method you must login first using the 'Login' web method.		
<b>Input Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Filename**	The name of the file that was uploaded	Include the file extension.
Content**	The contents of the file as a single string.	
EmailAddress**	The email address where the results of the upload is sent.	

### Output parameters:

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.

## 2.13 “UploadCRNLookupZipFile” web method

The UploadCRNLookupZipFile API method is designed to give merchants the ability to upload a CRN lookup zip file onto the server. Note once a CRN lookup file has been uploaded all previous CRN’s from this merchant is deleted.

### Input parameters:

<b>Authentication Parameters</b>		
There are no authentication parameters, but in order to use this method you must login first using the ‘Login’ web method.		
<b>Input Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Filename**	The name of the file that was uploaded	Include the file extension.
Content**	The content of the file as a single byte stream.	
EmailAddress**	The email address where the results of the upload is sent.	

### Output parameters:

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.

## 2.14 “SubmitBatch” web method

The SubmitBatch API method is designed to give merchants the ability to upload and immediately process a valid batch file.

### Input parameters:

<b>Authentication Parameters</b>		
There are no authentication parameters, but in order to use this method you must login first using the 'Login' web method.		
<b>Input Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Filename**	The name of the file that was uploaded including the extension.	Include the file extension.
Content**	The content of the file as a single string.	

### Output parameters:

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
<b>Output Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
errors	A list of all the problems with the submitted batch.	If the returned list is empty, then the batch file was uploaded and processed successfully.



## 2.15 “RetrieveBatchStatistics” web method

The SubmitBatch API method is designed to give merchants the ability to view all batches that were uploaded in a given time period.

### Input parameters:

<b>Authentication Parameters</b>		
There are no authentication parameters, but in order to use this method you must login first using the ‘Login’ web method.		
<b>Input Parameters</b>		
Parameter name	Description	Comments
timePeriod**	0 = Last 24 hours (default) 1 = last 7 days 2 = last 30 days 3 = last 45 days	

### Output parameters:

<b>Web Service Response Parameters</b>		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
<b>Output Parameters</b>		
Parameter name	Description	Comments
BatchFileID	ID of the batch file	
BatchState	State the batch is currently in	
Completed Time	Time the batch completed processing the transaction.	If batch has not been processed, then this string is empty
OrigFilename	Filename of the batch	
Percentage	Percentage of the batch processed.	Is the value ProcCount / TxnCount
ProcCount	Number of transactions that have been processed	
ResFilename	Filename of the response file	
SubmittedTime	Time the batch was submitted for processing.	
TxnCount	Number of transactions in the batch file.	

## 2.16 “DownloadBatchFileByFilename” web method

The DownloadBatchFileByFilename API method is designed to give merchants the ability to download any processed batch file from the server.

### Input parameters:

<b>Authentication Parameters</b>		
There are no authentication parameters, but in order to use this method you must login first using the 'Login' web method.		
<b>Input Parameters</b>		
Parameter name	Description	Comments
filename**	The filename of the batch that will be searched for.	
fileFormat**	1 = BPoint 2 = VirtualPOS 3 = VirtualPOSPlus	
returnAll**	True = returns both approved and declined transactions False = returns only declined transactions	

### Output parameters:

<b>Web Service Response Parameters</b>		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
<b>Output Parameters</b>		
Parameter name	Description	Comments
Content	The contents of the batch file as a byte stream	
Filename	The filename of the batch that was returned	
MerchantNumber	The merchant number of the batch that was returned.	This is the merchant number of the merchant who uploaded the file.

## 2.17 “DownloadBatchFileByID” web method

The DownloadBatchFileByID API method is designed to give merchants the ability to download any processed batch file from the server.

### Input parameters:

<b>Authentication Parameters</b>		
There are no authentication parameters, but in order to use this method you must login first using the 'Login' web method.		
<b>Input Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
BatchField**	The batchfileID of the batch that will be searched for.	
fileFormat**	1 = BPoint 2 = VirtualPOS 3 = VirtualPOSPlus	
returnAll**	True = returns both approved and declined transactions False = returns only declined transactions	

### Output parameters:

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
<b>Output Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Content	The contents of the batch file as a byte stream	
Filename	The filename of the batch that was returned	
MerchantNumber	The merchant number of the batch that was returned.	This is the merchant number of the merchant who uploaded the file.

## 2.18 “RetrieveSettlementFiles” web method

The RetrieveSettlementFiles API method is designed to give merchants the ability to download the settlement file for any given date(s).

### Input parameters:

<b>Authentication Parameters</b>		
There are no authentication parameters, but in order to use this method you must login first using the ‘Login’ web method.		
<b>Input Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
settlementDate**	The date of the settlement reports to be retrieved.	The date should be in format YYYYMMDD.

### Output parameters:

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
<b>Output Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Content	The contents of the batch file as a byte stream	
Filename	The filename of the settlement report.	

## 2.19 “Login” web method

The Login API method must be used to login in order to invoke some methods. You may need to configure your cookies to maintain the state between the web client and the web service.

Authentication Parameters		
Parameter name	Description	Comments
cliVersion**	The Command Line Interface	The parameter is just a reference, and only used for reporting purposes.
username**	Username provided to you	
password**	Password provided to you	
Merchant Number**	Your merchant facility number	

### Output parameters:

Web Service Response Parameters		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
Output Parameters		
Parameter name	Description	Comments
isLoggedIn	True = Successfully logged in False = Not logged in	

## 2.20 “Logout” web method

The Login API method is used to logout.

### Output parameters:

Web Service Response Parameters		
Parameter name	Description	Comments
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.

## 2.21 “GetAnonymousToken” web method

The GetAnonymousToken API method should be invoked after successful redirection to merchant’s web site from Add Anonymous Data Vault Token page. This method verifies that parameters passed in redirection have not been tampered with and returns the details of the token that was created for the specified CRNs.

### Input parameters:

<b>Authentication Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
<b>Input Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
success	Success parameter passed in redirection.	
CRN1	Value of parameter CRN1 passed in redirection.	
CRN2	Value of parameter CRN2 passed in redirection.	
CRN3	Value of parameter CRN3 passed in redirection.	
signature	Signature passed in redirection.	

### Output parameters:

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for “SUCCESS” before proceeding with any other checks	If “ERROR” is returned then “ResponseMessage” will contain an appropriate error message.
<b>Output Parameters (VerificationResult)</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
VerificationResult	CardNumber	
<b>Output Parameters (DVTOKEN)</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
CardNumber	Credit card number for the relating token.	For security reasons will always be null.
CardType	The card type used to process the payment	Eg. “MC” = mastercard “VC” = visacard
CreatedDate	Date the token was created	

CreatedUsername	Username of the person that created the token	
CRN1	It is an additional reference	
CRN2	Additional reference	
CRN3	Additional reference	
ExpiryDate	The expiry date on card	
MaskedCardNumber	The first 6 and last 3 digits of the credit card number charged returned back to you	
Token	Token Number	
UpdatedDate	Date last updated	
UpdatedUsername	Username of the person that last updated the token	

## 2.22 “TokenisePayment” web method

The TokenisePayment API method allows tokenisation of a card number for a particular transaction if the customer agreed to allow storage of the card number and the transaction was approved. This method should be invoked after successful redirection to merchant’s web site from Payment Connector payment. This method returns the details of the token that was created.

### Input parameters:

Authentication Parameters		
Parameter name	Description	Comments
Username	Username provided to you	This is a compulsory parameter
Password	Password provided to you	This is a compulsory parameter
Merchant number	Your merchant facility number	This is a compulsory parameter
Input Parameters		
Parameter name	Description	Comments
txnNumber	Unique transaction identifier that the merchant would have received in a redirection from Payment Connector to merchant receipt page. The parameter name in redirection is: <b>out_txn_number</b>	

**Output parameters:**

<b>Web Service Response Parameters</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
ResponseCode	You will need to check this for "SUCCESS" before proceeding with any other checks	If "ERROR" is returned then "ResponseMessage" will contain an appropriate error message.
<b>Output Parameters (DVToken)</b>		
<b>Parameter name</b>	<b>Description</b>	<b>Comments</b>
CardNumber	Credit card number for the relating token.	For security reasons will always be null.
CardType	The card type used to process the payment	Eg. "MC" = mastercard "VC" = visacard
CreatedDate	Date the token was created	
CreatedUsername	Username of the person that created the token	
CRN1	The CRN values will be the same as used in the transaction that was tokenised.	
CRN2		
CRN3		
ExpiryDate	The expiry date on card	
MaskedCardNumber	The first 6 and last 3 digits of the credit card number charged returned back to you	
Token	Token Number	
UpdatedDate	Date last updated	
UpdatedUsername	Username of the person that last updated the token	



---

## 3 Code Samples

---

### 3.1 C-Sharp (C#)

To create a web reference in Visual Studio, right click the web reference folder in the solution explorer and select 'Add Web Reference'.

Add the following web reference:

<https://www.bpoint.com.au/evolve/service.asmx>

You may give the web reference any name, in the following examples the web reference is named 'bPointAPI'.

#### Example 1 - Process Payment

```
//Create the service
bPointAPI.Service bPointSVC = new bPointAPI.Service();

//Create a transaction request
bPointAPI.TxnRequest trans = new bPointAPI.TxnRequest();
trans.Amount = 1000;
trans.CardNumber = "5123456789012346";
trans.CRN1 = "Tester";
trans.CRN2 = "";
trans.CRN3 = "";
trans.CVC = "123";
trans.ExpiryDate = "0513";
trans.MerchantReference = "";
trans.OriginalTransactionNumber = "";
trans.PaymentType = bPointAPI.PaymentType.PAYMENT;
trans.TxnType = bPointAPI.TxnType.INTERNET_ANONYMOUS;

bPointAPI.ServiceResponse resp;

//Webcall to process the payment
bPointAPI.TxnResponse pay = bPointSVC.ProcessPayment("username",
"password", "0000000000000001", trans, out resp);

//Check the result
if (resp.ResponseCode == bPointAPI.ResponseCode.SUCCESS)
    string result = pay.AuthorisationResult;
```

## Example 2 - Retrieve Batch Statistics

```
//Create the service
bPointAPI.Service bPointSVC = new bPointAPI.Service();

bPointAPI.ServiceResponse resp;

//Login
bPointSvc.CookieContainer = New System.Net.CookieContainer();
bool loggedin = newsvc.Login("1.2.2.489", "username", "password",
"0000000000000001", out resp);

//Webcall to retrieve batchstats
PointAPI.BatchStats[] listofbatches =
bPointSVC.RetrieveBatchStatistics(1, out resp);

//Check the result
if (resp.ResponseCode == bPointAPI.ResponseCode.SUCCESS)
{
    foreach (bPointAPI.BatchStats bs in listofbatches)
    {
        string batchname = bs.OrigFilename;
        /*Process batchstats*/
    }
}
```

## 3.2 Visual Basic (VB.NET)

To create a web reference in Visual Studio, right click the web reference folder in the solution explorer and select 'Add Web Reference'.

Add the following web reference:

<https://www.bpoint.com.au/evolve/service.asmx>

You may give the web reference any name, in the following examples the web reference is named 'bPointAPI'.

### Example 1 - Process Payment

```
'Create the service
Dim bPointSvc As bPointAPI.Service = New bPointAPI.Service()

'Create a transaction request
Dim trans As bPointAPI.TxnRequest = New bPointAPI.TxnRequest()
trans.Amount = 1000
trans.CardNumber = "5123456789012346"
trans.CRN1 = "Tester"
trans.CRN2 = ""
trans.CRN3 = ""
trans.CVC = "123"
trans.ExpiryDate = "0513"
trans.MerchantReference = ""
trans.OriginalTransactionNumber = ""
trans.PaymentType = bPointAPI.PaymentType.PAYMENT
trans.TxnType = bPointAPI.TxnType.INTERNET_ANONYMOUS

Dim resp As bPointAPI.ServiceResponse = New
bPointAPI.ServiceResponse()

'Webcall to process the payment
Dim pay As bPointAPI.TxnResponse =
bPointSvc.ProcessPayment("apitest", "gE0NJU05o8", "0000000000000001",
trans, resp)

'Check the result
If (resp.ResponseCode = bPointAPI.ResponseCode.SUCCESS) Then
    Dim result As String = pay.AuthorisationResult
End If
```

## Example 2 - Retrieve Batch Statistics

```
'Create the service
Dim bPointSvc As bPointAPI.Service = New bPointAPI.Service()

Dim resp As bPointAPI.ServiceResponse = New
bPointAPI.ServiceResponse()

'Login
bPointSvc.CookieContainer = New System.Net.CookieContainer()
bPointSvc.Login("1.2.2.489", "username", "password",
"0000000000000001", resp)

'Webcall to retrieve batchstats
Dim listofbatches As bPointAPI.BatchStats() =
bPointSvc.RetrieveBatchStatistics(1, resp)

'Check the result
If (resp.ResponseCode = bPointAPI.ResponseCode.SUCCESS) Then
    Dim bs As bPointAPI.BatchStats

    For Each bs In listofbatches
        Dim batchname As String = bs.OrigFilename
        'Process payment
    Next bs
End If
```

### 3.3 PHP

To be able to connect to the webservice make sure the following PHP extensions are installed and enabled on your PHP server:

- PHP\_OpenSSL, required to connect to a secure URL
- PHP\_SOAP, required to create the service

#### Example 1 - Process Payment

```
//Create the client
$client = new
SoapClient("https://www.bpoint.com.au/evolve/service.asmx?WSDL");

//Create a transaction request
$trans = array(
    "Amount"      => 1000,
    "CardNumber" => "5123456789012346",
    "CRN1"        => "Tester",
    "CRN2"        => "",
    "CRN3"        => "",
    "CVC"         => "123",
    "ExpiryDate" => "0513",
    "PaymentType" => "PAYMENT",
    "TxnType"     => "INTERNET_ANONYMOUS",
    "MerchantReference" => "OnlineBikeStore",
    "OriginalTransactionNumber" => "");

//Webcall to process the payment
$pay = $client->ProcessPayment(array('username' =>"myusername",
'password' => "mypassword", 'merchantNumber' => "0000000000000000",
'txnReq' => $trans));

//Check the result
If ($pay->response->ResponseCode == 'SUCCESS')
    echo (string)$pay->ProcessPaymentResult->AuthorisationResult;
```

## Example 2 – Transaction Search

```
//Create the client
$client = new
SoapClient("https://www.bpoint.com.au/evolve/service.asmx?WSDL");

//Create the request
$stransearch = array(
    "MerchantNumber" => "",
    "PaymentType" => "NOT_SPECIFIED",
    "TxnType" => "NOT_SPECIFIED",
    "TxnType" => "0",
    "NumOfHours" => "0",
    "FromDate" => "2010-10-01T13:00:01",
    "ToDate" => "2010-12-14T20:10:01",
    "ReceiptNumber" => "",
    "MerchantReference" => "",
    "CRN1" => "",
    "CRN2" => "",
    "CRN3" => "",
    "Amount" => "",
    "MaskedCardNumber" => "",
    "ExpiryDate" => "",
    "ResponseCode" => "",
    "AcquirerResponseCode" => "",
    "TransactionNumber" => "",
    "AuthoriseId" => "",
    "SettlementDate" => "",
    "CardType" => "",
    "BillerCode" => "",
    "SearchOrder" => "ASCENDING");

//Webcall to search the transactions
$search = $client->SearchTransactions(array('username'
=>"myusername", 'password' => "mypassword", 'merchantNumber' =>
"0000000000000000", 'search' => $stransearch));

//Check the result
If ($search->response->ResponseCode == 'SUCCESS') {
    echo 'Success';
    //Process the search results
}
Else
    echo $search->response->ResponseMessage;
```

## 3.4 Java

Open up NetBeans IDE

Create a new Java Web Application

- Select File -> New Project -> Java Web -> Web Application, click next
- Give project name, click Finish

Add new Web Service Client

- Right Click on Project Solution
- Select New -> web service client
- In the popup wizard, select WSDL URL and add in the following URL  
<https://www.bpoint.com.au/evolve/service.asmx?WSDL>
- Enter Package Name as **BpointWebService**
- Tick the box for "Generate Dispatch Code", click Finish
- Accept certificate

Add new Client Servlet to Consume Web Service

- Right Click on Project Solution
- Select New -> Servlet
- In the popup wizard, Enter Name of servlet and Enter Package Name as clientservlet, click Next
- Tick the box "Add information to deployment descriptor", click Finish

Open up the servlet

Add import statements

- import BPointWebService.PaymentType;
- import BPointWebService.TxnType;
- import javax.xml.ws.WebServiceRef;

Add the following code below the beginning of the class for the web reference

- @WebServiceRef(wsdlLocation = "WEB-INF/wsdl/evolve-  
uat.premier.com.au/newevolveservice/service.asmx.wsdl")
- private BPointWebService.Service service;

### Example 1 - Process Payment

```
package clientservlet;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

// Add import statements
import javax.xml.ws.WebServiceRef;
import BPointWebService.PaymentType;
import BPointWebService.TxnType;

public class ProcessPayment extends HttpServlet {
```

```
// Add web reference to WSDL
@WebServiceRef(wsdlLocation =
"WEBINF/wsdl/bpoint.com.au/evolve/service.asmx.wsdl")
private BPointWebService.Service service;

protected void processRequest(HttpServletRequest request,
HttpServletRequest response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();

//Create new BPoint Soap web service
BPointWebService.ServiceSoap port = service.getServiceSoap();

try {
// Create new txn Request Object for the payment transaction
BPointWebService.TxnRequest txnReq = new
BPointWebService.TxnRequest();
    txnReq.setAmount(1000);
    txnReq.setCardNumber("5123456789012346");
    txnReq.setCRN1("tester");
    txnReq.setCRN2("");
    txnReq.setCRN3("");
    txnReq.setCVC("123");
    txnReq.setExpiryDate("0513");
    txnReq.setMerchantReference("");
    txnReq.setOriginalTransactionNumber("");
    txnReq.setPaymentType(PaymentType.PAYMENT);
    txnReq.setTxnType(TxnType.INTERNET_ANONYMOUS);

// Create Response Objects
javax.xml.ws.Holder<BPointWebService.TxnResponse>
processPaymentResponse = new
javax.xml.ws.Holder<BPointWebService.TxnResponse>();
    javax.xml.ws.Holder<BPointWebService.ServiceResponse>
serviceResponse = new
javax.xml.ws.Holder<BPointWebService.ServiceResponse>();

// Consume Process Payment method on web service
port.processPayment("username", "password", "0000000000000000",
txnReq, processPaymentResponse, serviceResponse);

// Output payment response

out.println(processPaymentResponse.value.getResponseCode());
out.println(processPaymentResponse.value.getAcquirerResponseCode());
out.println(processPaymentResponse.value.getAuthorisationResult());
out.println(processPaymentResponse.value.getTransactionNumber());
out.println(processPaymentResponse.value.getReceiptNumber());
out.println(processPaymentResponse.value.getAuthoriseId());
out.println(processPaymentResponse.value.getSettlementDate());
out.println(processPaymentResponse.value.getMaskedCardNumber());
out.println(processPaymentResponse.value.getCardType());
```



```
// Output service response (SUCCESS or FAIL)

out.println(serviceResponse.value.getResponseCode());
out.println(serviceResponse.value.getResponseMessage());

} catch (Exception ex) {
    out.println(ex.toString());
}

out.close();
}
```

## Example 2 - Retrieve Batch Statistics

NOTE: You must allow Web Service Session Cookies, this code is highlighted in red

```
package clientservlet;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

// Add import statements
import javax.xml.ws.BindingProvider;
import javax.xml.ws.WebServiceRef;

public class RetrieveBatchStatistics extends HttpServlet {
// Add web reference to WSDL
@WebServiceRef(wsdlLocation = "WEB-INF/wsdl/evolve-
uat.premier.com.au/newevolveservice/service.asmx.wsdl")
private BPointWebService.Service service;

protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();

//Create new BPoint Soap web service
BPointWebService.ServiceSoap port = service.getServiceSoap();

// Enable Web Session Cookies
((BindingProvider)port).getRequestContext().put(BindingProvider.SESSI
ON_MAINTAIN_PROPERTY, true);

try {
// Create Response Objects
javax.xml.ws.Holder<java.lang.Boolean> loginResponse = new
javax.xml.ws.Holder<java.lang.Boolean>();
javax.xml.ws.Holder<BPointWebService.ServiceResponse>
serviceResponse = new
javax.xml.ws.Holder<BPointWebService.ServiceResponse>();
// Login to Bpoint through web service
```

```
port.login("1.2.2.489", "username", "password",
"0000000000000001", loginResponse, serviceResponse);

// Check login successful
if(loginResponse.value.booleanValue() == true)
{

    javax.xml.ws.Holder<BPointWebService.ArrayOfBatchStats>
    batchStatsArray = new
    javax.xml.ws.Holder<BPointWebService.ArrayOfBatchStats>();

    // Consume Retrieve Batch Statistics method on web service
    port.retrieveBatchStatistics(1, batchStatsArray,
    serviceResponse);

    if(serviceResponse.value.getResponseCode().toString().equals("S
    UCCCESS"))
    {
        // Output Batch Statistics
        java.util.List<BPointWebService.BatchStats> batchStats =
        batchStatsArray.value.getBatchStats();
        for(int i = 0; i < batchStats.size(); i++)
        {
            BPointWebService.BatchStats bStat =
            batchStats.get(i);
            out.println(bStat.getBatchFileId());
            out.println(bStat.getTxnCount());
            out.println(bStat.getProcCount());
            out.println(bStat.getOrigFilename());
            out.println(bStat.getResFilename());
            out.println(bStat.getSubmittedTime());
            out.println(bStat.getCompletedTime());
            out.println(bStat.getBatchState());
            out.println(bStat.getPercentage());
        }
    }

    // Output service response (SUCCESS or FAIL)
    out.println(serviceResponse.value.getResponseCode());

    out.println(serviceResponse.value.getResponseMessage());
}
else
{

    out.println(serviceResponse.value.getResponseMessage());
}

} catch (Exception ex) {
    out.println(ex.toString());
}

    out.close();
}
```

## 4 Appendix

### 4.1 Transaction Response Codes

Response Code:	Issuer Response Code:	Description:
<b>Bank Response Codes:</b>		
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
<b>Gateway response codes:</b>		
?		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
<b>Payment Server Response Codes</b>		
?		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 CVC.
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.
-------	--	--------------------------------