

# Merchant Warrior Fuse

## Transparent Redirect Integration Guide

Last Updated - 01/2012



## Table of Contents

<b>PREAMBLE</b> .....	<b>3</b>
<b>RELATED DOCUMENTS</b> .....	<b>3</b>
<b>WHO SHOULD READ THIS DOCUMENT?</b> .....	<b>3</b>
<b>CONTACT</b> .....	<b>3</b>
<b>GENERAL INFORMATION</b> .....	<b>4</b>
INTRODUCTION .....	4
AVAILABLE METHODS .....	4
REQUEST FORMAT .....	4
RESPONSE FORMAT .....	4
<b>PERFORMING A PURCHASE</b> .....	<b>5</b>
REQUEST PARAMETER OVERVIEW .....	5
REQUEST PARAMETER TABLE - PROCESSCARD .....	6
REQUEST PARAMETER TABLE - PROCESSCARD (CONT) .....	7
REQUEST PARAMETER TABLE - PROCESSCARD (CONT) .....	8
RESPONSE.....	9
<b>PERFORMING A PRE-AUTHORIZATION</b> .....	<b>11</b>
REQUEST PARAMETER OVERVIEW .....	11
REQUEST PARAMETER TABLE - PROCESSAUTH.....	12
REQUEST PARAMETER TABLE - PROCESSAUTH (CONT) .....	13
REQUEST PARAMETER TABLE - PROCESSAUTH (CONT) .....	14
RESPONSE.....	15
<b>PERFORMING A REFUND OR CAPTURE</b> .....	<b>16</b>
<b>STORING CARD DATA</b> .....	<b>17</b>
REQUEST PARAMETER OVERVIEW .....	17
REQUEST PARAMETER TABLE - ADDCARD .....	18
RESPONSE.....	19
<b>APPENDIX A – GENERATING THE VERIFICATION HASH</b> .....	<b>21</b>
OVERVIEW.....	21
TRANSACTION TYPE HASH .....	21
URL HASH .....	21
302 REDIRECT & POST NOTIFICATION VERIFICATION HASH .....	21
CUSTOM FIELDS VERIFICATION HASH .....	21

## Preamble

**Merchant Warrior** (MW) is an Australian based payment provider that offers a range of online payment solutions to Merchants worldwide. MW enables Merchants to connect to the major banks throughout Australia, and via our international partners, banks throughout Europe, American and Asia.

MW is a PCI DSS Certified Tier 1 gateway. This qualification was obtained after an independent audit carried out by Securus Global, a certified QSA and QPSAC.

## Related Documents

Please also see the document entitled “Warrior Express – Integration Guide”, which should be used as an accompaniment. It contains information on transaction verification, as well as information relating to the asynchronous POST notification, appendixes, etc. As the MWF platform leverages the existing MWE API, all of the information in that document relating to transaction processing is pertinent, and has been intentionally left out of this document to prevent duplication.

## Who should read this document?

This document has been written to assist the programmers responsible for integrating MWF into a third party codebase. The assumption has been made that the reader will be familiar with basic industry standard terms such as HTTP, SSL, POST, XML, etc, and that they will have a basic understanding of the way that a hosted sales page generally works.

## Contact

If anything in this document is unclear, incorrect or requires clarification, please don't hesitate to contact our support department, either by phone on **+617 3166 5489** or via email at [infoservices@merchantwarrior.com](mailto:infoservices@merchantwarrior.com). Our support department team is available 24/7.

## General Information

The following sub-sections of this document will outline the various API methods present in the **Merchant Warrior Fuse** (MWF) solution.

### Introduction

MWF, using the MWE API provided by Merchant Warrior, allows a Merchant to create and maintain a single integration point, regardless of the banking partner (provider) they choose to use.

Before you are able to connect to the MWF platform, you must first obtain a Merchant ID, API Key and API Passphrase. These will be issued to you once you create a MWF account.

### Available Methods

The MWF platform consists of a hosted page on Merchant Warrior's servers that can be utilized as a frontend to the MWE API, as well as a "Transparent Redirect" option that allows the Merchant to host the payment form, but not capture the payment details. Two types of transaction are available through MWF: purchase and pre-auth/capture. For purchase, please refer to processCard method, and for pre-auth/capture, please refer to processAuth and processCapture methods.

The ability to store credit card details in the Merchant Warrior Vault (Token Payments) is also available through the Transparent Redirect via the addCard method.

This document outlines the Transparent Redirect method of connectivity.

### Request Format

All requests are sent to the MWF solution using the HTTP POST method, and must be performed over the HTTPS protocol. A payment form is hosted on the Merchant's website, which has an action of one of the endpoints listed below.

For the Transparent Redirect version of MWF, the requests must be sent to either <https://api.merchantwarrior.com/transfer/> (live) or <https://base.merchantwarrior.com/transfer/> (testing).

### Response Format

All MWF responses, regardless of the status, are returned via asynchronous POST as text/xml and contain, **at minimum**, the following XML elements:

```
<mwResponse>  
  <responseCode></responseCode>  
  <responseMessage></responseMessage>  
</mwResponse>
```

The element "mwResponse" will contain (at least) two child elements – responseCode and responseMessage. responseCode will be an signed integer (see Appendix D – Response Code Mapping) which allows the Merchant to quickly identify the status of the response. responseMessage will contain a textual string which offers more information on the specific response received.

Along with the POST notification, a 302 Redirect is returned to the Merchant via the client's browser once a transaction is completed. This request contains a query string, which contains the Transaction ID for querying, as well as summary data (transaction state, etc).

## Performing a Purchase

In order to perform a purchase transaction, a form must be posted to one of the request endpoints listed above, with a number of required (and some optional) fields. Assuming the request validates correctly, the transaction will be processed and the result will be returned to the Merchant.

### Request Parameter Overview

The following parameters are accepted. Parameters that are **bold** are required.

```
method
merchantUUID
apiKey
transactionAmount
transactionCurrency
transactionProduct
transactionReferenceID
returnURL
notifyURL
urlHash
hashSalt
customerName
customerCountry
customerState
customerCity
customerAddress
customerPostCode
customerPhone
customerEmail
customerIP
paymentCardNumber
paymentCardExpiry
paymentCardName
paymentCardCSC
addCard
custom1
custom2
custom3
hash
```

For a more details about each field, please refer to the table on the next page.

## Request Parameter Table - processCard

Required	Parameter
<b>API Method</b>	
Y	method
	<b>Example:</b> processCard <b>Notes:</b> This is case sensitive.
<b>Authentication Parameters</b>	
Y	merchantUUID
	<b>Example:</b> 123456789abcd <b>Notes:</b> The value of this parameter is assigned to you by Merchant Warrior.
Y	apiKey
	<b>Example:</b> 1a3b5c <b>Notes:</b> The value of this parameter is assigned to you by Merchant Warrior.
<b>General Transaction Parameters</b>	
Y	transactionAmount
	<b>Example:</b> 10.00 <b>Notes:</b> The amount must be formatted to have two decimal places. Any amounts without two decimal places or amounts less than one cent will be rejected.
Y	transactionCurrency
	<b>Example:</b> AUD <b>Notes:</b> One of the following: AUD, CAD, EUR, GBP, JPY, NZD, SGD, USD. This is provider dependant. Please check with MW before attempting to process transactions in any currency other than AUD. This field is case insensitive.
Y	transactionProduct
	<b>Example:</b> Test Product <b>Notes:</b> A product (or sale) description. This field's primary purpose is to help the transaction be identifiable for reporting and accounting purposes. <b>Valid Length:</b> Up to 255 characters. Some banks limit this field to 40 characters.
N	transactionReferenceID
	<b>Example:</b> A257240023321 <b>Notes:</b> This is merchant's unique reference ID for a transaction sent to Merchant Warrior. The main purpose of this ID is to verify the transaction via queryCard method in the event a valid response is not received. <b>Valid Length:</b> Up to 16 characters.
<b>Redirect &amp; Notification Parameters</b>	
Y	returnURL
	<b>Example:</b> <a href="https://www.example.com/return.php">https://www.example.com/return.php</a> <b>Notes:</b> The customer will be redirected to this URL upon completion of the transaction
Y	notifyURL
	<b>Example:</b> <a href="https://www.example.com/notify.php">https://www.example.com/notify.php</a> <b>Notes:</b> The asynchronous POST notifications will be sent to this URL.
Y	urlHash
	<b>Example:</b> 511999e54b9ad51ce4c28d7f0550ac81 <b>Notes:</b> The urlHash field is a combination of the MD5 of your API Passphrase, and specific parameters sent in the transaction. See Appendix A – Generating the Verification Hash (Batch URL Hash) for information on how to construct the hash correctly. <b>Valid Length:</b> 32 characters.

## Request Parameter Table - processCard (cont)

Required	Parameter
Y	hashSalt  <b>Example:</b> 3x4mpl3s4lt! <b>Notes:</b> Used to salt the return hash used in the 302 Redirect to redirectURL upon the completion of a transaction.
<b>Customer Parameters</b>	
Y	customerName  <b>Example:</b> Mr. Example Person <b>Notes:</b> This field can only contain alphanumeric characters, as well as the full stop and hyphen character. This field <b>MUST</b> be the full name of the customer and <b>MUST</b> contain at least one space character. <b>Valid Length:</b> Between 3 and 255 characters.
Y	customerCountry  <b>Example:</b> AU <b>Notes:</b> Two letter ISO 3166-1 alpha-2 country code. <b>Valid Length:</b> Two characters.
Y	customerState  <b>Example:</b> Queensland <b>Notes:</b> Freeform field, keep consistent for your records and reporting. <b>Valid Length:</b> Up to 75 characters.
Y	customerCity  <b>Example:</b> Brisbane <b>Notes:</b> Freeform field, keep consistent for your records and reporting. <b>Valid Length:</b> Up to 75 characters.
Y	customerName  <b>Example:</b> Mr. Example Person <b>Notes:</b> This field can only contain alphanumeric characters, as well as the full stop and hyphen character. This field <b>MUST</b> be the full name of the customer and <b>MUST</b> contain at least one space character. <b>Valid Length:</b> Between 3 and 255 characters.
Y	customerAddress  <b>Example:</b> 123 Test Street <b>Notes:</b> Freeform field. <b>Valid Length:</b> Up to 255 characters.
Y	customerPostcode  <b>Example:</b> 4000 <b>Notes:</b> This can also accommodate ZIP/Post codes for international transactions. <b>Valid Length:</b> Between 4 and 10 characters.
N	customerPhone  <b>Example:</b> 0401234567 or 61731234567 <b>Notes:</b> Anything other than +, -, space and 0-9 will be stripped. <b>Valid Length:</b> Up to 25 characters.
N	customerEmail  <b>Example:</b> <a href="mailto:person@example.com">person@example.com</a> <b>Notes:</b> Must be valid if present. Sending this optional parameter is highly recommended. <b>Valid Length:</b> Up to 255 characters.
N	customerIP  <b>Example:</b> 123.456.789.012 or 2001:0db8:85a3:0000:0000:8a2e:0370:7334 <b>Notes:</b> Any valid IPv4 or IPv6 address is accepted. Sending this optional parameter is highly recommended. <b>Valid Length:</b> Up to 39 characters.

## Request Parameter Table - processCard (cont)

Required	Parameter
<b>Payment Parameters</b>	
Y	<p>paymentCardNumber</p> <p><b>Example:</b> 5123456789012346 or 4557012345678902</p> <p><b>Notes:</b> Only certain card numbers are deemed valid in the test environment. See Appendix C – Test Data for more information. Do not send separators with the card number (e.g. 1234-5678... or 1234 5678).</p> <p><b>Valid Length:</b> Between 13 and 16 digits.</p>
Y	<p>paymentCardExpiry</p> <p><b>Example:</b> 0513</p> <p><b>Notes:</b> This must be in MMY format. The month must be zero-padded if it's less than 10.</p> <p><b>Valid Length:</b> 4 digits.</p>
Y	<p>paymentCardName</p> <p><b>Example:</b> Mr. Example Person or MR E PERSON or Example Person</p> <p><b>Notes:</b> This must contain at the very least a space and no less than two characters. Only alphanumeric characters, hyphens, spaces and full stops are allowed.</p> <p><b>Valid Length:</b> Between 3 and 255 characters.</p>
N	<p>paymentCardCSC</p> <p><b>Example:</b> 123</p> <p><b>Notes:</b> This is also known as the CVN or CVV/2. This is required if you have the feature enabled. Please contact Merchant Warrior for more information.</p>
N	<p>addCard</p> <p><b>Example:</b> 1</p> <p><b>Notes:</b> This value is a boolean to denote whether the paymentCardNumber should automatically be added to the Merchant Warrior Token System after processing the transaction.</p> <p><b>Valid Length:</b> 1 digit.</p>
<b>Custom Fields</b>	
Y	<p>custom1</p> <p><b>Example:</b> Custom Field 1</p> <p><b>Notes:</b> Freeform field. Returned as &lt;custom1&gt; in the XML response.</p> <p><b>Valid Length:</b> Up to 500 characters.</p>
Y	<p>custom2</p> <p><b>Example:</b> Custom Field 2</p> <p><b>Notes:</b> Freeform field. Returned as &lt;custom2&gt; in the XML response.</p> <p><b>Valid Length:</b> Up to 500 characters.</p>
Y	<p>custom3</p> <p><b>Example:</b> Custom Field 3</p> <p><b>Notes:</b> Freeform field. Returned as &lt;custom3&gt; in the XML response.</p> <p><b>Valid Length:</b> Up to 500 characters.</p>
Required	Parameter
<b>Verification Hash</b>	
Y	<p>hash</p> <p><b>Example:</b> e9ddc296b76b3398934bfc06239073df</p> <p><b>Notes:</b> The verification hash is a combination of the MD5 of your API Passphrase, and specific parameters sent in the transaction. See Appendix A – Generating the Verification Hash (Transaction Type Hash) for information on how to construct the hash correctly.</p> <p><b>Valid Length:</b> 32 characters.</p>

## Response

### Asynchronous POST

As stated in the “Response Format” section, all responses for the methods outlined in this document follow the same format, and are sent to the notifyURL via asynchronous HTTP POST. Take the following for example:

#### Successful Transaction:

(assuming addcard = 0 – credit card details will not be stored)

```
<mwResponse>
  <responseCode>0</responseCode>
  <responseMessage>Transaction approved</responseMessage>
  <transactionID>1-918490ae-9a1c-11de-8649-000c29753ad4</transactionID>
  <authCode>1</authCode>
  <authMessage>Approved</authMessage>
  <authResponseCode>0</authResponseCode>
  <custom1>Custom Field 1</custom1>
  <custom2>Custom Field 2</custom2>
  <custom3>Custom Field 3</custom3>
  <customHash>e1af7c6a17758ef2907a50c96fa56d28</customHash>
  <hash>082949a8dfaccda185a135db425377b</hash>
</mwResponse>
```

#### Successful Transaction:

(assuming addcard = 1 – credit card details will be stored)

```
<mwResponse>
  <responseCode>0</responseCode>
  <responseMessage>Transaction approved</responseMessage>
  <transactionID>1-918490ae-9a1c-11de-8649-000c29753ad4</transactionID>
  <authCode>1</authCode>
  <authMessage>Approved</authMessage>
  <authResponseCode>0</authResponseCode>
  <custom1>Custom Field 1</custom1>
  <custom2>Custom Field 2</custom2>
  <custom3>Custom Field 3</custom3>
  <customHash>e1af7c6a17758ef2907a50c96fa56d28</customHash>
  <cardID>243</cardID>
  <cardKey>fd3Az0m0zMT51Ft2</cardKey>
  <hash>082949a8dfaccda185a135db425377b</hash>
</mwResponse>
```

If, however, the transaction did not pass MWF's validation (due to an incorrect parameter, etc), a response similar to the following will be returned:

#### Failed Response:

```
<mwResponse>
  <responseCode>-1</responseCode>
  <responseMessage>MW - 008:Invalid merchantUUID / apiKey
  combination</responseMessage>
</mwResponse>
```

The <responseCode> and <responseMessage> fields will **always** be present in the transaction response. There are three possible types of <responseCode>'s that can be returned:

1. <responseCode> < 0. MWF validation error.
2. <responseCode> = 0. Transaction was successful
3. <responseCode> > 0. Transaction was declined by the provider.

If the transaction passed MWF's validation, then generally the <auth\*> fields will be present in the response. The <auth\*> fields contain the upstream provider response data.

Please refer to "Warrior Express – Integration Guide" for more information and sample code.

## Performing a Pre-authorization

In order to perform a pre-authorization transaction, a form must be posted to one of the request endpoints listed above, with a number of required (and some optional) fields. Assuming the request validates correctly, the transaction will be processed and the result will be returned to the Merchant.

### Request Parameter Overview

The following parameters are accepted. Parameters that are **bold** are required.

```
method
merchantUUID
apiKey
transactionAmount
transactionCurrency
transactionProduct
transactionReferenceID
returnURL
notifyURL
urlHash
hashSalt
customerName
customerCountry
customerState
customerCity
customerAddress
customerPostCode
customerPhone
customerEmail
customerIP
paymentCardNumber
paymentCardExpiry
paymentCardName
paymentCardCSC
addCard
custom1
custom2
custom3
hash
```

For a more details about each field, please refer to the table on the next page.

## Request Parameter Table - processAuth

Required	Parameter
<b>API Method</b>	
Y	method
	<p><b>Example:</b> processAuth</p> <p><b>Notes:</b> This is case sensitive.</p>
<b>Authentication Parameters</b>	
Y	merchantUUID
	<p><b>Example:</b> 123456789abcd</p> <p><b>Notes:</b> The value of this parameter is assigned to you by Merchant Warrior.</p>
Y	apiKey
	<p><b>Example:</b> 1a3b5c</p> <p><b>Notes:</b> The value of this parameter is assigned to you by Merchant Warrior.</p>
<b>General Transaction Parameters</b>	
Y	transactionAmount
	<p><b>Example:</b> 10.00</p> <p><b>Notes:</b> The amount must be formatted to have two decimal places. Any amounts without two decimal places or amounts less than one cent will be rejected.</p>
Y	transactionCurrency
	<p><b>Example:</b> AUD</p> <p><b>Notes:</b> One of the following: AUD, CAD, EUR, GBP, JPY, NZD, SGD, USD. This is provider dependant. Please check with MW before attempting to process transactions in any currency other than AUD. This field is case insensitive.</p>
Y	transactionProduct
	<p><b>Example:</b> Test Product</p> <p><b>Notes:</b> A product (or sale) description. This field's primary purpose is to help the transaction be identifiable for reporting and accounting purposes.</p> <p><b>Valid Length:</b> Up to 255 characters. Some banks limit this field to 40 characters.</p>
N	transactionReferenceID
	<p><b>Example:</b> A257240023321</p> <p><b>Notes:</b> This is merchant's unique reference ID for a transaction sent to Merchant Warrior. The main purpose of this ID is to verify the transaction via queryCard method in the event a valid response is not received.</p> <p><b>Valid Length:</b> Up to 16 characters.</p>
<b>Redirect &amp; Notification Parameters</b>	
Y	returnURL
	<p><b>Example:</b> <a href="https://www.example.com/return.php">https://www.example.com/return.php</a></p> <p><b>Notes:</b> The customer will be redirected to this URL upon completion of the transaction</p>
Y	notifyURL
	<p><b>Example:</b> <a href="https://www.example.com/notify.php">https://www.example.com/notify.php</a></p> <p><b>Notes:</b> The asynchronous POST notifications will be sent to this URL.</p>
Y	urlHash
	<p><b>Example:</b> 511999e54b9ad51ce4c28d7f0550ac81</p> <p><b>Notes:</b> The urlHash field is a combination of the MD5 of your API Passphrase, and specific parameters sent in the transaction. See Appendix A – Generating the Verification Hash (Batch URL Hash) for information on how to construct the hash correctly.</p> <p><b>Valid Length:</b> 32 characters.</p>

## Request Parameter Table - processAuth (cont)

Required	Parameter
Y	hashSalt <b>Example:</b> 3x4mpl3s4lt! <b>Notes:</b> Used to salt the return hash used in the 302 Redirect to redirectURL upon the completion of a transaction.
<b>Customer Parameters</b>	
Y	customerName <b>Example:</b> Mr. Example Person <b>Notes:</b> This field can only contain alphanumeric characters, as well as the full stop and hyphen character. This field <b>MUST</b> be the full name of the customer and <b>MUST</b> contain at least one space character. <b>Valid Length:</b> Between 3 and 255 characters.
Y	customerCountry <b>Example:</b> AU <b>Notes:</b> Two letter ISO 3166-1 alpha-2 country code. <b>Valid Length:</b> Two characters.
Y	customerState <b>Example:</b> Queensland <b>Notes:</b> Freeform field, keep consistent for your records and reporting. <b>Valid Length:</b> Up to 75 characters.
Y	customerCity <b>Example:</b> Brisbane <b>Notes:</b> Freeform field, keep consistent for your records and reporting. <b>Valid Length:</b> Up to 75 characters.
Y	customerName <b>Example:</b> Mr. Example Person <b>Notes:</b> This field can only contain alphanumeric characters, as well as the full stop and hyphen character. This field <b>MUST</b> be the full name of the customer and <b>MUST</b> contain at least one space character. <b>Valid Length:</b> Between 3 and 255 characters.
Y	customerAddress <b>Example:</b> 123 Test Street <b>Notes:</b> Freeform field. <b>Valid Length:</b> Up to 255 characters.
Y	customerPostcode <b>Example:</b> 4000 <b>Notes:</b> This can also accommodate ZIP/Post codes for international transactions. <b>Valid Length:</b> Between 4 and 10 characters.
N	customerPhone <b>Example:</b> 0401234567 or 61731234567 <b>Notes:</b> Anything other than +, -, space and 0-9 will be stripped. <b>Valid Length:</b> Up to 25 characters.
N	customerEmail <b>Example:</b> <a href="mailto:person@example.com">person@example.com</a> <b>Notes:</b> Must be valid if present. Sending this optional parameter is highly recommended. <b>Valid Length:</b> Up to 255 characters.
N	customerIP <b>Example:</b> 123.456.789.012 or 2001:0db8:85a3:0000:0000:8a2e:0370:7334 <b>Notes:</b> Any valid IPv4 or IPv6 address is accepted. Sending this optional parameter is highly recommended. <b>Valid Length:</b> Up to 39 characters.

## Request Parameter Table - processAuth (cont)

Required	Parameter
<b>Payment Parameters</b>	
Y	<p>paymentCardNumber</p> <p><b>Example:</b> 5123456789012346 or 4557012345678902</p> <p><b>Notes:</b> Only certain card numbers are deemed valid in the test environment. See Appendix C – Test Data for more information. Do not send separators with the card number (e.g. 1234-5678... or 1234 5678).</p> <p><b>Valid Length:</b> Between 13 and 16 digits.</p>
Y	<p>paymentCardExpiry</p> <p><b>Example:</b> 0513</p> <p><b>Notes:</b> This must be in MMY format. The month must be zero-padded if it's less than 10.</p> <p><b>Valid Length:</b> 4 digits.</p>
Y	<p>paymentCardName</p> <p><b>Example:</b> Mr. Example Person or MR E PERSON or Example Person</p> <p><b>Notes:</b> This must contain at the very least a space and no less than two characters. Only alphanumeric characters, hyphens, spaces and full stops are allowed.</p> <p><b>Valid Length:</b> Between 3 and 255 characters.</p>
N	<p>paymentCardCSC</p> <p><b>Example:</b> 123</p> <p><b>Notes:</b> This is also known as the CVN or CVV/2. This is required if you have the feature enabled. Please contact Merchant Warrior for more information.</p> <p><b>Valid Length:</b> Between 3 and 4 characters.</p>
N	<p>addCard</p> <p><b>Example:</b> 1</p> <p><b>Notes:</b> This value is a boolean to denote whether the paymentCardNumber should automatically be added to the Merchant Warrior Token System after processing the transaction.</p> <p><b>Valid Length:</b> 1 digit.</p>
<b>Custom Fields</b>	
Y	<p>custom1</p> <p><b>Example:</b> Custom Field 1</p> <p><b>Notes:</b> Freeform field. Returned as &lt;custom1&gt; in the XML response.</p> <p><b>Valid Length:</b> Up to 500 characters.</p>
Y	<p>custom2</p> <p><b>Example:</b> Custom Field 2</p> <p><b>Notes:</b> Freeform field. Returned as &lt;custom2&gt; in the XML response.</p> <p><b>Valid Length:</b> Up to 500 characters.</p>
Y	<p>custom3</p> <p><b>Example:</b> Custom Field 3</p> <p><b>Notes:</b> Freeform field. Returned as &lt;custom3&gt; in the XML response.</p> <p><b>Valid Length:</b> Up to 500 characters.</p>
Required	Parameter
<b>Verification Hash</b>	
Y	<p>hash</p> <p><b>Example:</b> e9ddc296b76b3398934bfc06239073df</p> <p><b>Notes:</b> The verification hash is a combination of the MD5 of your API Passphrase, and specific parameters sent in the transaction. See Appendix A – Generating the Verification Hash (Transaction Type Hash) for information on how to construct the hash correctly.</p> <p><b>Valid Length:</b> 32 characters.</p>

## Response

Please refer to the Response section for the processCard method, as the responses are the same for all methods.

## Performing a Refund or Capture

Please refer to the “Warrior Express API” documentation for integration information regarding the *refundCard* and *processCapture* methods.

## Storing Card Data

In order to add a customer's credit card information to the Merchant Warrior Vault (Token Payments), a form must be posted to one of the requested endpoints listed above with a number of required (and some optional) fields. Assuming the request validates correctly, the credit card number will be stored (encrypted and securely) successfully and the result will be returned to the Merchant.

### Request Parameter Overview

The addCard method **requires** the following parameters.

```
merchantUUID  
apiKey  
cardName  
cardNumber  
cardExpiryMonth  
cardExpiryYear  
returnURL  
notifyURL  
urlHash  
hashSalt
```

For a more details about each field, please refer to the table on the next page.

## Request Parameter Table - addCard

Required	Parameter
<b>Authentication Parameters</b>	
Y	merchantUUID <b>Example:</b> 123456789abcd <b>Notes:</b> The value of this parameter is assigned to you by Merchant Warrior.
Y	apiKey <b>Example:</b> 1a3b5c <b>Notes:</b> The value of this parameter is assigned to you by Merchant Warrior.
<b>Cardholder Data</b>	
Y	cardName <b>Example:</b> Mr. Example Person or MR E PERSON or Example Person <b>Notes:</b> This must contain at the very least a space and no less than two characters. Only alphanumeric characters, hyphens, spaces and full stops are allowed. <b>Valid Length:</b> Between 3 and 255 characters.
Y	cardNumber <b>Example:</b> 5123456789012346 or 4557012345678902 <b>Valid Length:</b> Between 13 and 16 digits.
Y	cardExpiryMonth <b>Example:</b> 05 <b>Notes:</b> This must be in MM format. The month must be zero-padded if it's less than 10. <b>Valid Length:</b> 2 digits.
Y	cardExpiryYear <b>Example:</b> 13 <b>Notes:</b> This must be in YY format. <b>Valid Length:</b> 2 digits.
<b>Redirect &amp; Notification Parameters</b>	
Y	returnUrl <b>Example:</b> <a href="https://www.example.com/return.php">https://www.example.com/return.php</a> <b>Notes:</b> The customer will be redirected to this URL upon completion of the transaction
Y	notifyURL <b>Example:</b> <a href="https://www.example.com/notify.php">https://www.example.com/notify.php</a> <b>Notes:</b> The asynchronous POST notifications will be sent to this URL.
Y	urlHash <b>Example:</b> 511999e54b9ad51ce4c28d7f0550ac81 <b>Notes:</b> The urlHash field is a combination of the MD5 of your API Passphrase, and specific parameters sent in the transaction. See Appendix A – Generating the Verification Hash (Batch URL Hash) for information on how to construct the hash correctly. <b>Valid Length:</b> 32 characters.
Y	hashSalt <b>Example:</b> 3x4mpl3s4lt! <b>Notes:</b> Used to salt the return hash used in the 302 Redirect to returnUrl upon the completion of a transaction.

## Response

As stated in the “Response Format” section, all responses for the methods outlined in this API Document follow the same format. There may be extra fields for the Token Payment methods though, which will be illustrated per method. Take the following for example:

```
<mwResponse>
  <responseCode>0</responseCode>
  <responseMessage>Operation Successful</responseMessage>
  <cardID>1</cardID>
  <cardKey>a84JI2cA12ziZ3Fx</cardKey>
  <hash>c59bba3c09d66b4c1a934a1a33b55161</hash>
</mwResponse>
```

The addCard method will return a cardID and cardKey assuming the operation was successful. These values will need to be stored locally by the Merchant, as they’re used for all subsequent requests related to the added card data.

**NOTE: After the initial addCard via the Transparent Redirect, all future Token Payment operations (removal of card, update of expiry etc.) should be processed through the Warrior Express (Direct API) platform.**

The <responseCode> fields will **always** be present in the transaction response. There are three possible types of <responseCode>’s that can be returned:

1. <responseCode> < 0. MWV validation error.
2. <responseCode> = 0. Command was successful
3. <responseCode> > 0. Command failed due to non-validation error

For more information on the possible codes, please refer to the “Warrior Express API” documentation - *Appendix D - Response Code Mapping*.

### 302 Redirect

As an asynchronous notification is not always reliable, an immediate response is also sent in the form of a 302 redirect to the returnURL. For security reasons, not all of the transactional data is present in the result's Query string. The following table lists the possible return fields.

Field	Description
status	A textual representation of the transaction result.
reference	If status is 'error', this will not be present. Otherwise, this contains a transactionID that can be used to query the transaction via an API call with the queryCard method.
hash	A hash used to verify the status & transactionID (or cardID), using hashSalt. Refer to Appendix A.
Code	If status is 'error', this will contain an MWE error code.
message	If status is 'error', this will contain a textual representation of the error code.
custom1	The value of the <custom1> parameter, URL Encoded. Will be blank if not provided.
custom2	The value of the <custom2> parameter, URL Encoded. Will be blank if not provided..
custom3	The value of the <custom3> parameter, URL Encoded. Will be blank if not provided..
customHash	A hash used to verify the custom* parameters. Refer to Appendix A.

## Appendix A – Generating the Verification Hash

### Overview

The verification hash is used to prove to MWF that the transaction request being sent has been generated by the Merchant, and not a malicious third party who has discovered the Merchant's merchantUUID and apiKey. Even if somebody captures the POST data that gets sent to MWF, they will not be able to create new (or refund old) transactions without knowing the API Passphrase – which you are able to change whenever you want via the Merchant Warrior Barracks.

To generate a verification hash, you need to concatenate specific fields that you're sending MWF in a specific order, convert them to lower case, and then MD5 them.

### Transaction Type Hash

To generate a transaction type hash, concatenate the following fields:

```
md5(apiPassphrase) + merchantUUID + transactionAmount + transactionCurrency
```

Once concatenated, convert everything to lowercase, and then md5 the string:

```
Step 1 (concatenate):  
md5(passphrase)123456789abcd10.00AUD
```

```
Step 2 (convert to lower):  
md5(passphrase)123456789abcd10.00aud
```

```
Step 3 (md5):  
d941117d8774b12e218650542af6af56
```

### URL Hash

To generate a URL hash, concatenate the following fields:

```
md5(apiPassphrase) + merchantUUID + returnUrl + notifyURL
```

Once concatenated, convert everything to lowercase, and then md5 the string, as above.

### 302 Redirect & POST Notification Verification hash

To generate a verification hash used for the 302 Redirect and POST notification, concatenate the following fields:

```
md5(apiPassphrase) + hashSalt + merchantUUID + status + transactionID (or  
cardID)
```

**NOTE: cardID is used for verifying addCard response.**

The status and transactionID (or cardID) fields are both contained in the Redirect URL. Once concatenated, convert everything to lowercase, and then md5 the string, as above.

### Custom Fields Verification hash

To generate the custom fields hash, concatenate, convert to lowercase & md5 the following fields:

```
md5(apiPassphrase) + custom1 + custom2 + custom3
```

Be sure to decode the custom\* fields first – e.g. "Custom+Field+1" becomes "Custom Field 1".