



PaywayWS Credit Card Integration Guide

Date: September 17, 2020

NOTICE:

This technical specification is the confidential and proprietary product of Payway®, Inc. Any unauthorized use, reproduction, or transfer of this specification is strictly prohibited. All rights reserved.

Copyright © 2021 by Payway®, Inc.

Table of Contents

Revision History	7
Related Documentation	8
Introduction	9
I. Integration Process	10
A. Development current (paywaywsdev.com)--deprecated (devedgilpayway.net)	10
B. Staging current (paywaywstest.com)—deprecated (testedgilpayway.net)	10
C. Production current (paywayws.com)—deprecated (edgilpayway.com,edgilpayway.net)	10
II. Server Locations	12
III. RESTful Resources	12
IV. Background vs. Interactive Requests	13
V. Sale vs Authorize / Capture	15
VI. Payway Account Tokens	15
VII. Background Requests	16
A. Get A Payway Session	16
B. Trial Authorize	17
C. Authorize Request	17
D. Capture Authorized Transaction	18
E. Sale Request	19
F. Sale Pre Authorized	19
G. Credit (Refund)	21
H. Void (Cancel)	21
I. Add Account	22
J. Read Transaction Information	22
K. Read Account Information	23
L. Update Account Data	24
M. Update Transaction Notes	25

VIII. Interactive Requests	26
A. Queue Sale Transaction	27
B. Queue Trial Authorize	28
C. Queue Authorize Transaction	29
D. Queue Credit Transaction	30
E. Send Queued Transaction	31
F. Get Hosted Transaction Results	32
G. Queue Add Account Request	33
H. Send Queued Add Account Request	34
I. Get Hosted Add Account Results	34
J. getApplePaySession	36
IX. Testing Your Integration	37
X. JSON Request Fields	38
A. Top Level	38
B. Card Account	41
C. Card Transaction	43
XI. JSON Reply Fields	47
A. Top Level	47
B. CardAccount	47
C. CardTransaction	49
XII. Payway Result Codes (paywayCode field)	54
XIII. Processor Specific Result Codes	66
A. TSYS	66
B. First Data Merchant Services and Chase Paymentech	69
XIV. Request / Reply Examples	71
A. Get A Payway Session	71
B. Trial Authorize with Primary Account Number	72
C. Authorize with Primary Account Number	75
D. Sale with Primary Account Number	78

E. Pre Authorized Sale with Primary Account Number	82
F. Credit With Primary Account Number	85
G. Sale with Payway Account Token	88
H. Trial Authorize with Payway Account Token	91
I. Pre Authorized Sale with Payway Account Token	93
J. Credit with Payway Account Token	96
K. Authorize with Payway Account Token	99
L. Sale with P2PE Data	102
M. Credit with P2PE Data	106
N. Authorize with P2PE Data	110
O. Capture Authorized Transaction	114
P. Void Transaction	117
Q. Add New Credit Card with Primary Account Data	119
R. Add New Account With P2PE Data	122
S. Retrieve Account Data	124
T. Retrieve Transaction Data	126
U. Update Account Data	128
V. Update Transaction Notes	129
W. Queue Authorize Hosted Request	131
X. Queue Sale Hosted Request	133
Y. Queue Trial Authorize	134
Z. Send A Queued Transaction	135
AA. Retrieve Hosted Transaction Results	137
BB. Queue Add Account Hosted Request	140
CC. Send Queued Add Account	141
DD. Retrieve Hosted Add Account Results	142

XV. Payway Test Response Results.

143

Revision History

Revision	Date	Initials	Comments
527	07/02/2020	DRN	Initial Draft
529	09/17/2020	DRF	Added Support for updateAccount page 16,86
532	2/1/2021	DRF	Corrected JSON reply message from 'state' to 'status' with the correct integer values on page 36.
533	3/1/2021	DRF	Added updateTransactionNotes method.
533	3/1/2021	DRF	Updated and standardized convention for: idSource is now sourceId for all references. idDivision is now divisionId for all references. idCompany is now companyId for all references. Note the old convention is backwards compatible EXCEPT new methods added after release 532.
533	3/1/2021	DRF	Changed all references to 'settled' to 'deposited' in comment explanations.
533	3/1/2021	DRF	The 'status' table on page 40 was updated so that a status of 4 is Pending and status of 5 is Deposited.
533	5/20/2021	DRF	Added 'authorize request with primary account data' JSON example in Section XIV, B.
533	5/21/2021	DRF	Updated payway URLs for development, test and production access in Sections I,II.
534	6/3/2021	DRF	Added method 'trialAuthorize" for verification requests in section VII, B.
534	6/2/2021	DRF	Added method queueTrialAuthorize for verification requests in Section VIII, B.
534	6/2/2021	DRF	Added JSON request types 'trialAuthorize" and queueTrialAuthorize" to Section X, A.

534	6/2/2021	DRF	Added JSON examples for trialAuthorize and queueTrialAuthorize in sections XIV, Section B,H,Y.
534	6/22/2021	DRF	Clarified soft descriptor use and warning in section X, C page 39.
534	6/22/2021	DRF	Added Section XV, Payway Test Response Results.
534	6/30/2021	DRF	Added status 'Verified' on page 43.
534	6/30/2021	DRF	Added 'type' JSON element with descriptions on page 45.
534	6/30/2021	DRF	Added missing type reply in the card-transaction element to all JSON examples.

Related Documentation

- PaywayWS ACH Integration Guide
- PaywayWS Redirected Payment Form Integration Guide
- Payway® Users Guide
- Payway® Admin Guide
- Payway® ApplePay Integration Guide
- Payway® GooglePay Integration Guide
- Payway® Point-to-Point Encryption(P2PE) User's Guide
- Payway® File Processing Users Guide

Introduction

PaywayWS is a RESTful web service providing services related to credit card payments, ACH payments and account management.

This document assumes that you have a working knowledge of payment processing gateways, RESTful web services, and JSON data structures.

I. Integration Process

Your integration with PaywayWS is a three-stage process. During each of the integration stages described, you will have unlimited access to our integration development system and technical support.

You will be assigned a Payway technical expert who will help with any questions on technical- and payment-related issues.

Once in production, you can get support by calling 800-457-9932 option 1 or emailing support@payway.com.

A. Development current (paywaywsdev.com)--deprecated (devedgilpayway.net)

We provide a development sandbox where Payway's development and support teams can review your integration. This will allow us to monitor your requests and responses in order to quickly help you debug your integration.

B. Staging current (paywaywstest.com)—deprecated (testedgilpayway.net)

Once your integration is debugged, you will move to the Payway staging system to allow you to run your regression testing and acceptance processing. The staging system is a duplicate of the production system, except you will use different logon credentials.

We will ask you to submit a set of transactions that will help to make sure the incoming data is correct and that error conditions are being handled. The validation is customized to your integration so that you will only be concerned with PaywayWS features that you plan to use in production. We will provide immediate turnaround on your validation transactions so there will be minimal lag time in your development/deployment cycle.

C. Production current (paywayws.com)—deprecated

(edgilpayway.com,edgilpayway.net)

Before going live, you will receive credentials to our production system. Validation testing on our production system is done in coordination with Payway support. This is an end-to-end test that includes verification of live deposits into each of your merchant accounts.

II. Server Locations

The RESTful application name is PaywayWS. The hosted sites are listed below:

Development Server: <https://paywaywsdev.com>

Staging Server: <https://paywaywstest.com>

Production Server: <https://paywayws.com>

Example, to process a payment transaction on the development system the URL would be: <https://paywaywsdev.com/PaywayWS/Payment/CreditCard>.

III. RESTful Resources

The following table describes each of the RESTful resources supported by PaywayWS. The PATH column gives the path of the resource. The host specification and application name are not shown in the table.

PATH	DESCRIPTION
/Session	Creates all types of sessions including Payway user sessions, Apple Pay® sessions, and others.
/Payment/CreditCard	Create and manage transactions for credit card payment types.
/Account/CreditCard	Create and manage credit card accounts including adding an account and acquiring the related token.
/Query/CreditCard	Retrieve detailed results for one or more credit card transactions or credit card accounts. Sensitive data is not returned or is returned masked.

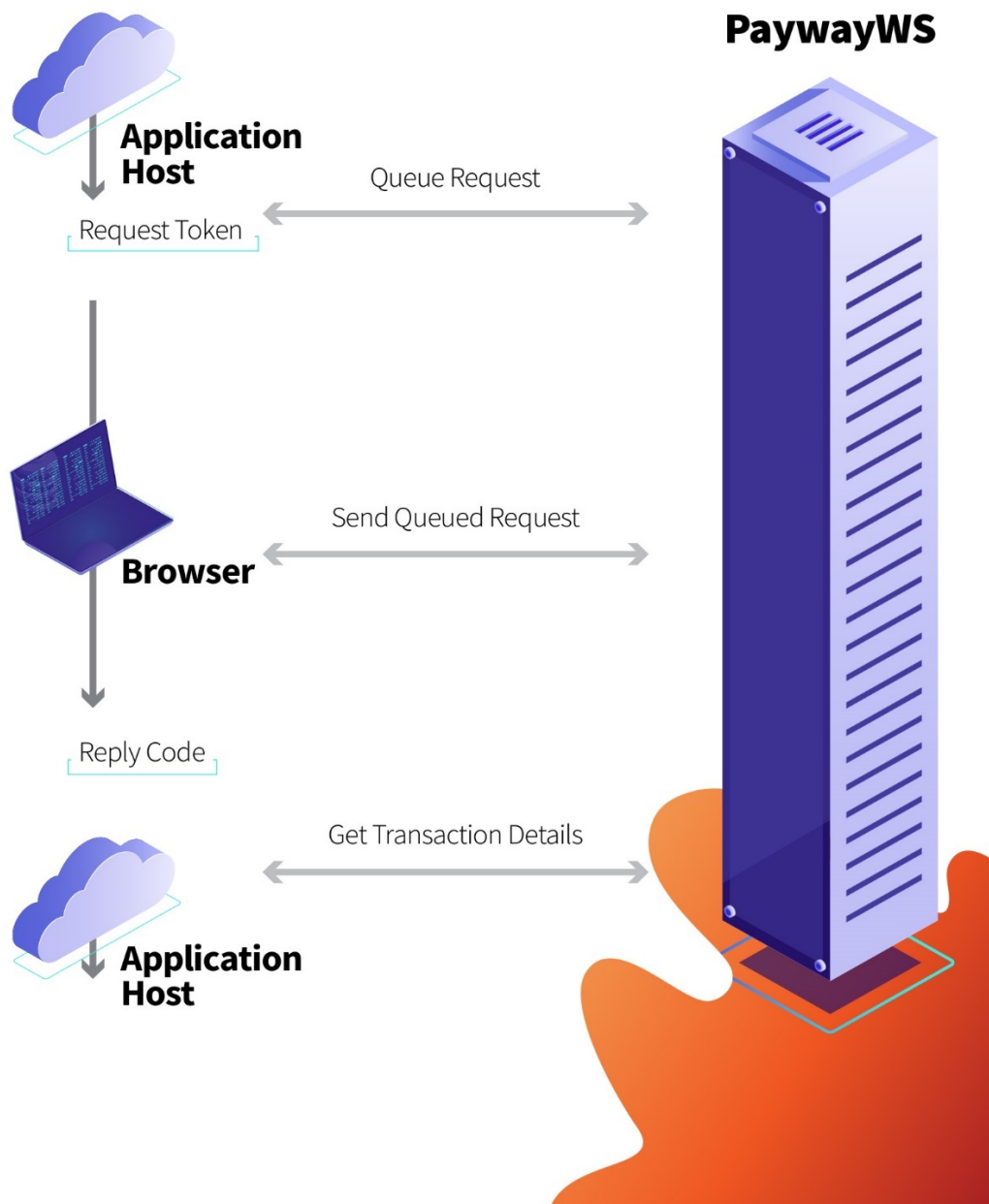
IV. Background vs. Interactive Requests

Background Requests are requests that originate on your server directly to PaywayWS. Generally, these are recurring transactions where you already have a Payway account token. A common use case for background requests is to execute a batch of Payway account token-based recurring payments. It is possible to do a background request with full card data, but there are PCI validation ramifications to this.

Interactive requests are used when a web user is involved in processing a transaction, for example purchasing a good or service from your web site. Interactive requests are always done in three steps, queue the request, send the request, and process the results. Requests are queued from your server. Then, when the user submits the payment on the browser, a java script is called to send the previously-queued request. The result code is then posted to your server for processing. Optionally you can query transaction details when the result code is returned. This pattern is followed for the implementation of Hosted Payment Entry, Hosted Account Entry, and all of the wallets (ApplePay, GooglePay, etc.).

The following diagram shows the interactive (web) request pattern in general.

PaywayWS Interactive Request Pattern



V. Sale vs Authorize / Capture

There are two methods for capturing funds from a cardholder's account. The first is to authorize, which puts a hold on the account for the amount authorized, followed by a capture, which causes the held amount to be transferred to the merchant's account. The second method is a sale request, which is equal to an authorize and capture in a single step.

Authorize / capture is used in a fulfillment environment where the funds must be held but the sale is not finalized until the product or service is delivered. If an Authorize is not followed by a Capture request, the hold on the funds will eventually expire, but it is strongly suggested that you execute a void against any authorization that will not be captured.

Sale is used when the charge will be made immediately, for example, when processing a recurring subscription charge. **NOTE: If a sale is requested but the results are not available for whatever reason, do not request another sale until you request the data for the original request and inspect it to make sure it has not been captured. Otherwise double charges may occur. One way to avoid this is to make sure you create unique transaction IDs. Then, if you send the same transaction ID in, PaywayWS will reply with a duplicate notice and the status of the already-existing transaction.**

VI. Payway Account Tokens

For both interactive and background requests, a Payway account token is returned in the final result. This may be a new token referring to a newly-created account, or a pre-existing token. These account tokens can be stored and used for any request related to the account that is referred to by the token.

There is no need to add an account explicitly using addAccount to generate a token. If you send a sale or other type of request with account data, a token referring to the new account—possibly an existing account if all the account data matches an existing account—is returned in the detailed results.

VII. Background Requests

See the Examples section below for samples of all requests.

A. Get A Payway Session

JSON 'request' value	getPaywaySession
DESCRIPTION	Obtains a Payway session token based upon presented credentials
URL	/Session
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	<p>Most PaywayWS requests require a valid Payway session token, which must be retrieved using this request. The exception is sendQueuedTransaction which requires a paywayRequestToken. Session tokens can be used for multiple requests. Session tokens that are not used for 5 minutes are deleted and a new session will have to be requested.</p> <p>Multiple active sessions for the same user are allowed.</p> <p>DO NOT CALL THIS METHOD FROM A BROWSER. This is to be called from a secure site connected to PaywayWS.</p> <p>User credentials for the development, testing, and live systems will be provided by Payway for integration and setup.</p> <p>The Payway_Admin web application is used by merchants to manage users and credentials.</p>

B. Trial Authorize

JSON 'request' value	trialAuthorize
DESCRIPTION	trialAuthorize performs a verification of \$0.00 on the cardholder's credit card. No credit hold is issued on the cardholder's account. The accountInputMode can either be primaryAccountNumber or paywayAccountToken. If primaryAccountNumber is used, payway will also return a vault token(paywayAccountToken) in the JSON Reply that corresponds to the customer's account. This can be used in future subsequent transactions.
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	This feature must be enabled by payway prior to production requests.

C. Authorize Request

JSON 'request' value	Authorize
DESCRIPTION	Authorize puts a hold, for the amount of the authorization, on the cardholder's account. A subsequent capture request must be executed to transfer the funds.
URL	/Payment/CreditCard
METHOD	Post

URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

D. Capture Authorized Transaction

JSON 'request' value	capture
DESCRIPTION	Capture previously authorized funds
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	Transactions that are authorized place a hold on the account holder's account for the amount of the authorization. These funds can be captured by calling the method described below. Funds will be transferred during the daily settlement process. Separate Authorize and Capture is used in a fulfillment environment where the payment is authorized at purchase time and the goods or service are delivered later.

E. Sale Request

JSON 'request' value	sale
DESCRIPTION	A sale request is equivalent to an Authorize, followed by a Capture. A hold is placed on the credit card account, and funds are transferred during the nightly settlement.
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

F. Sale Pre Authorized

JSON 'request' value	salePreAuthorized
DESCRIPTION	In some cases, an authorization is performed outside of PaywayWS. In order to enter these transactions into Payway, the salePreAuthorized request is supported. The authorization code received externally is entered as part of this request. The funds are deposited in the same

	manner as authorizations performed through PaywayWS.
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

G. Credit (Refund)

JSON 'request' value	credit
DESCRIPTION	A credit request will transfer funds from the merchant account to the cardholder. This is often called a refund.
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

H. Void (Cancel)

JSON 'request' value	void
DESCRIPTION	Transactions that have not yet been deposited may be voided using the void request. After a transaction has been deposited (money transferred), a credit request is required to refund the money to the cardholder. Void will succeed on transactions that are authorized or captured, but not yet deposited. In general, voids must be done on the same day as the authorize, capture, or sale. Otherwise a refund must be issued.
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json

PRODUCES	text/json
NOTES	

I. Add Account

JSON 'request' value	addAccount
DESCRIPTION	This request is used to add a payment account to the PaywayWS vault for later use.
URL	/Account/CreditCard
METHOD	Put
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	A paywayAccountToken is returned that can be used in future transactions.

J. Read Transaction Information

JSON 'request' value	getTransaction
DESCRIPTION	Returns the transaction and payment account information for the specified transaction.
URL	/Query/CreditCard
METHOD	Get

URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	Transactions are uniquely identified by a source ID and transaction name.

K. Read Account Information

JSON 'request' value	getAccount
DESCRIPTION	Used to find a particular payment account.
URL	/Query/CreditCard
METHOD	Get
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	Payment accounts are uniquely identified by division ID and account token.

L. Update Account Data

Use this method to update account data. Account number cannot be updated. If there is a new account number use addAccount and you will get a new token. All other fields normally input by addAccount can be updated with this method, including expiration date. Note that if you are using Payway's Account Updater expiration dates will automatically be updated when they change

JSON 'request' value	updateAccount
DESCRIPTION	Used to update a particular payment account.
URL	/Query/CreditCard
METHOD	Put
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	Payment accounts are uniquely identified by division ID and account token.

M. Update Transaction Notes

Use this method to update optional notes fields for a specific transaction.

Only the notes fields can be updated.

JSON 'request' value	updateTransactionNotes
DESCRIPTION	Used to update notes for specific transaction.
URL	/Payment/CreditCard
METHOD	Put
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	Notes fields can only be updated.

VIII. Interactive Requests

Payway supports secure web-based transactions for your applications. Many payment methods are available for interactive transactions, including payment wallets, and hosted payment entry which redirects credit card data entry to PaywayWS and eliminates the need for your server to receive any sensitive credit card data.

To support these transactions PaywayWS provides the ability to queue transactions. This sets up the transaction on PaywayWS and returns to you a `paywayRequestToken`. This token is what is passed to the browser, and an associated java script is used to send the account information entered by the user plus the `transactionToken` to PaywayWS.

The `paywayRequestToken` can be used ONLY once and ONLY for the transaction previously queued from your server. The user has no ability to change amounts or other transaction attributes.

See the related documents section at the beginning of this document for more information on Hosted Payment Entry and various wallet payment methods offered by PaywayWS.

A. Queue Sale Transaction

JSON 'request' value	queueSale
DESCRIPTION	<p>The queueSale request does not require a Payway session token. Your credentials are passed in this request and the paywayRequestToken, a one-time use token, is returned.</p> <p>After queueSale the card page is loaded and sendQueuedTransaction is called via a java script.</p>
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

B. Queue Trial Authorize

JSON 'request' value	queueTrialAuthorize
DESCRIPTION	<p>queueTrialAuthorize request does not require a Payway session token. Credentials are passed in this request and the paywayRequestToken, a one-time use token, is returned along with the payway vault token corresponding to the customer's credit card. This method can be used to verify a card account as well as generated a payway token that can be used in subsequent transactions.</p> <p>A queueTrialAuthorize performs a verification of \$0.00 on the cardholder's credit card. No credit hold is issued on the cardholder's account.</p>
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	<p>This method can only be used where the accountInputMode is the primaryAccount Number.</p> <p>This feature must be enabled by payway prior to production requests.</p>

C. Queue Authorize Transaction

JSON 'request' value	queueAuthorize
DESCRIPTION	<p>The queueAuthorize request does not require a Payway session token. Your credentials are passed in this request and the paywayRequestToken, a one-time use token, is returned.</p> <p>After queueAuthorize the card page is loaded and sendQueuedTransaction is called via a java script.</p> <p>A background capture request must be executed on a successful authorize for the sale to be completed funds deposited.</p>
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

D. Queue Credit Transaction

JSON 'request' value	queueCredit
DESCRIPTION	The queueCredit request does not require a Payway session token. Your credentials are passed in this request and the paywayRequestToken, a one-time use token, is returned.
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

E. Send Queued Transaction

JSON 'request' value	sendQueuedTransaction
DESCRIPTION	<p>This request will send a previously queued transaction.</p> <p>Account information entered by the user is passed in this request.</p>
URL	/Payment/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	<p>The form of the account data depends on the payment method the user has chosen. It could be primary account data, Apple Pay encrypted data, Google Pay data, or P2PE encrypted data. This is sent from the browser via a java script when the user clicks on a particular button to make the payment.</p> <p>See the various wallet and payment method integration guides for more information.</p>

F. Get Hosted Transaction Results

JSON 'request' value	hostedTransactionResults
DESCRIPTION	The hostedTransactionResults request is executed from the browser and returns only the result code and a possible text error message.
URL	/Query/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	This request is executed from your server and returns all fields for the related transaction. Account sensitive data is masked, and internal fields used only by PaywayWS are not returned.

G. Queue Add Account Request

JSON 'request' value	queueAddAccount
DESCRIPTION	To allow interactive web users to add a payment account, you first have to set up a queueAddAccount request from your server to PaywayWS. This is similar to the queueSale type requests described above. A paywayRequestToken is returned to be sent to PaywayWs with the account information the users add.
URL	Account/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	If you are participating in the PaywayWS Account Updater program, any updates to the related account, such as a new expiration date, are applied to the original token. You do not have to acquire a new token for the updated account.

H. Send Queued Add Account Request

JSON 'request' value	sendQueuedAddAccount
DESCRIPTION	Called from a browser java script which redirects the primary account number input directly to PaywayWS.
URL	Account/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	The format of the data depends on the account input method.

I. Get Hosted Add Account Results

JSON 'request' value	hostedAddAccountResults
DESCRIPTION	Used in conjunction with queueAddAccount and sendAddAccount to execute a web-based hosted add account request. This request will return all fields for the related payment account.
URL	/Query/CreditCard
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	Sensitive data is masked, and internal fields used only by PaywayWS are not returned.

J. getApplePaySession

This request is called only from the ApplePay.js java script provided for integrations with PaywayWS ApplePay. See the ApplePay integration guide for more information.

JSON 'request' value	getApplePaySession
DESCRIPTION	Validates Apple pay merchant based upon your domain and returns Apple pay session data.
URL	/Session
METHOD	Post
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

IX. Testing Your Integration

Generate a test suite for your integration. PaywayWS provides a method to generate any of the PaywayWS error codes, and also to manage fraud security results, as well as address verification results.

The cardTransaction JSON object supports three request fields for testing various error conditions:

- `testResultCode` determines the reply code that will be returned from the test transaction. The reply code must be one of the values listed in Appendix I. If not, the return code will be 6000, PaywayWS internal error. If there is another actual error, such as account expired, this may override your requested return status.
- `testResultFSV` controls the contents of the returned `FraudSecurityResults` field. This is used for client-side processing of `FraudSecurityResults` or in testing the Payway Fraud Prevention component of PaywayWS. See the *Payway Fraud Prevention User's document* for more information.
- `testResultAVS` controls the contents of the returned `AddressVerificationResults`. This is used to test any processing of address verification results on the client side, or in testing the setting for the Payway Fraud Prevention component of PaywayWS. See the *Payway Fraud Prevention User's document* for more information.

X. JSON Request Fields

A. Top Level

Name	Description
request	<p>Request Name. This contains the request name, must be one of the following:</p> <p>“getPaywaySession”</p> <p>“getApplePaySession”</p> <p>“addAccount”</p> <p>“authorize”</p> <p>“trialAuthorize”</p> <p>“sale”</p> <p>“credit”</p> <p>“salePreAuthorized”</p> <p>“void”</p> <p>“capture”</p> <p>“getTransaction”</p> <p>“getAccount”</p> <p>“updateTransactionNotes”</p> <p>“queueSale”</p> <p>“queueAuthorize”</p> <p>“queueTrialAuthorize”</p> <p>“queueCredit”</p> <p>“sendQueuedTransaction”</p> <p>“hostedTransactionResults”</p> <p>“queueAddAccount”</p> <p>“sendQueuedAddAccount”</p> <p>“hostedAddAccountResults”</p>

userName	For getPaywaySession, Payway user name
password	For getPaywaySession, user's password
companyId	For getPaywaySession, company id for the user. This is part of the user configuration. See the PaywayAdmin application.
divisionId	For retrieving and adding data at the division level, for example the addAccount request.
accountInputMode	Specifies how the account information is added. Possible values are: "primaryAccountNumber" "magtek" "idtech"
paywaySessionToken	Used for all background requests except getPaywaySession. Session tokens expire after 5 minutes without being used.
paywayRequestToken	Used for interactive web-based requests. Returned by 'queue' type calls such as queueSale.
paywayAccountToken	Used for tokenized recurring transactions.
applePayToken	Used by the applePay java script. Encrypted Apple Pay data is decrypted on the PaywayWS server.
googlePayToken	Used by the googlePayway java script. Encrypted Google Pay data is decrypted by on the PaywayWS server.
transactionName	Used for requests requiring a transaction name and a source id. For example, capture and getTransaction.

sourceId	Used for requests requiring a transaction name and a source id. For example, capture and getTransaction.
----------	--

B. Card Account

Name	Description
firstName	First name from cardholder billing data
middleName	Middle name. Usually blank, not part of card processing
lastName	Last name from cardholder billing data
address	Address from cardholder billing data, may include unit number, apartment, etc.
city	City from cardholder billing data
state	State or province
zip	Zip code (US) or province code (CAD). It is strongly recommended to have at least zip available to reduce merchant fees and ideally as much of the cardholder name and address information as is available should be populated.
phone	Cardholder phone number. Optional, not part of card processing data.
accountNumber	Credit card account number. Required.
expirationDate	Expiration date in mmyy or mmyyyy format. Required.
p2peInput	For Payway P2PE devices, output of device goes in this field and is decrypted by Payway.

email	Email of the cardholder.
fsv	Fraud security value (3 digits or 4 digits for AMEX)
accountNotes1	Account notes, optional data up to 128 bytes
accountNotes2	Account notes, optional data up to 128 bytes
accountNotes3	Account notes, optional data up to 128 bytes

C. Card Transaction

Name	JSON Object	Description
name		<p>Unique name of the card transaction. Names must be unique within a division. If name field is empty, Payway will assign a unique transaction name that is returned with transaction replies.</p> <p>Required if not assigned by payway.</p>
amount		<p>Amount IN PENNIES. Ex \$1.23 submitted as '123'</p> <p>Required.</p>
tax		<p>Tax amount IN PENNIES. Ex \$1.23 submitted as '123.'</p> <p>Tax cannot be more than the 'amount.'</p>
eciType		<p>1 → MOTO Indicator/ Single Transaction mail/telephone order: designates a transaction where the cardholder is not present at a merchant location and consummates the sale via the phone or through the mail. The transaction is not for recurring services or product and does not include sales that are processed via an installment plan.</p> <p>2 → MOTO Indicator – Recurring Transaction: designates a transaction that represents an arrangement between a cardholder and the merchant where transactions are going to occur on a periodic basis. Required for all Card On File (COF) transactions including ApplePay and GooglePay.</p>

		<p>3 → MOTO Indicator – Installment Payment: designates a group of transactions that originated from a single purchase where the merchant agrees to bill the cardholder in installments.</p> <p>4 → MOTO Indicator – Deferred Transaction: designates a transaction that represents an order with a delayed payment for a specified amount of time.</p>
		<p>5 → ECI Indicator – Secure Electronic Commerce Transaction: designates a transaction between a cardholder and a merchant consummated via the Internet where the transaction was successfully authenticated and includes the management of a cardholder certificate. (e.g. 3-D Secure Transactions). <i>All original wallet transactions from ApplePay and GooglePay are an ECI of 5 EXCEPT VISA for ApplePay which is a 7. Note this value is supplied by GooglePay and ApplePay.</i></p> <p>6 → ECI Indicator – Non-Authenticated Electronic Commerce Transaction: designates a transaction consummated via the Internet at a 3-D Secure capable merchant that attempted to authenticate the cardholder using 3-D Secure. (e.g. 3-D Secure includes Verified by Visa and MasterCard SecureCode) Attempts occur with Verified by Visa and MasterCard SecureCode transactions in the event of:</p> <ul style="list-style-type: none"> a. A non-participating Issuer b. A non-participating cardholder of a participating Issuer

		<p>c. A participating Issuer, but the authentication server is not available</p> <p>7 → ECI Indicator – Channel Encrypted Transaction: designates a transaction between a cardholder and a merchant consummated via the Internet where the transaction includes the use of transaction encryption such as SSL, but authentication was not performed. The cardholder payment data was protected with a form of Internet security, such as SSL, but authentication was not performed. Generally, the first initial B2C transaction.</p> <p>8 → ECI Indicator – Non-Secure Electronic Commerce Transaction: designates a transaction between a cardholder and a merchant consummated via the Internet where the transaction does not include the use of any transaction encryption such as SSL, no authentication performed, no management of a cardholder certificate.</p> <p>R → Retail Indicator – designates a transaction where the cardholder was present at a merchant location. If an “R” is sent for a transaction with a MOTO Merchant Category Code (MCC), the transaction will downgrade.</p> <p>Required.</p>
processorSoftDescriptor		<p>Optional value, doing business as (DBA) description. Some merchant configurations must be pre-enabled for soft descriptors.</p> <p>Contact payway support for more information. Settlements will fail if this value is sent and the customer’s merchant</p>

		<p>is not configured to accept soft descriptors.</p> <p>This value will display on the consumer's card statement.</p> <p>Required for a recurring subscription transaction that is part of a discounted offer or package per VISA regulations.</p>
transactionNotes1		Transaction notes fields are optional, these can contain notes regarding the transaction, or an order or customer number to be used for tracking on Payway. Notes fields are searchable on Payway. Maximum length 64 bytes.
transactionNotes2		Transaction notes fields are optional. These can contain notes regarding the transaction, or an order or customer number to be used for tracking on Payway. Notes fields are searchable on Payway. Maximum length 32 bytes.
transactionNotes3		Transaction notes fields are optional, these can contain notes regarding the transaction, or an order or customer number to be used for tracking on Payway. Notes fields are searchable on Payway. Maximum length 32 bytes.
testResultAvs		For Testing only. This value will be set as the AVS Results upon return.
testResultFsv		For Testing only. This value will be set as the FSV Results upon return.

XI. JSON Reply Fields

A. Top Level

Name	Description
paywayCode	See the Payway Result Codes section below
paywayMessage	Human readable message. NOT TO BE PROCESSED.

B. CardAccount

All input fields are echoed back in the replies. For example, firstName, address. Below are additional reply fields that are not part of the input.

Name	Description										
paywayAccountToken	Account token used for recurring transactions										
divisionId	For retrieving and adding data at the division level, for example the addAccount request.										
status	<table><tr><th>Code</th><th>Description</th></tr><tr><td>2</td><td>Active</td></tr><tr><td>3</td><td>Replaced</td></tr><tr><td>4</td><td>Invalid</td></tr><tr><td>5</td><td>Closed</td></tr></table>	Code	Description	2	Active	3	Replaced	4	Invalid	5	Closed
Code	Description										
2	Active										
3	Replaced										
4	Invalid										
5	Closed										
lastUsed	Timestamp of the date the account was last used for a transaction.										
inputMode	Payway reserved field for internal processing.										

cardType	Code	Description
	1	Visa
	2	Master Card
	3	American Express
	5	Diners Club
	6	Discover
	8	Unknown
commercialCardType	Payway reserved field used for Level III processing.	
auLastUpdate	Timestamp of the date the account information was last updated for the cardholder given the token and new information. Previous versions of the cardholder data are saved in Payway for historical purposes.	
rawAuUpdateType	Unused	
firstFour	First four digits of the cardholder account number.	
lastFour	Last four digits of the cardholder account number.	

C. CardTransaction

All input fields are echoed back in the reply. Below are additional reply fields that are not part of the input. Other fields may be returned for informational purposes.

Name	Description																
sourceId	Each Transaction has a source id identifying where the transaction originated within your organization. Each division will be assigned one or more sources depending on your requirements. The source is used for reporting.																
name	Unique name for the transaction																
resultCode	An integer value representing the result of the request. See Section XII for a complete listing of all codes.																
status	<table> <tr> <th>Status</th><th>Description</th></tr> <tr> <td>3</td><td>Authorized : The transaction was authorized.</td></tr> <tr> <td>4</td><td>Pending: The transaction is waiting to be deposited.</td></tr> <tr> <td>5</td><td>Deposited : The transaction has been sent to be deposited.</td></tr> <tr> <td>6</td><td>Voided: The transaction has been voided.</td></tr> <tr> <td>7</td><td>Error: The transaction has been rejected due to an error</td></tr> <tr> <td>8</td><td>Declined: The transaction has been declined.</td></tr> <tr> <td>9</td><td>Verified: The Transaction has been verified.</td></tr> </table>	Status	Description	3	Authorized : The transaction was authorized.	4	Pending: The transaction is waiting to be deposited.	5	Deposited : The transaction has been sent to be deposited.	6	Voided: The transaction has been voided.	7	Error: The transaction has been rejected due to an error	8	Declined: The transaction has been declined.	9	Verified: The Transaction has been verified.
Status	Description																
3	Authorized : The transaction was authorized.																
4	Pending: The transaction is waiting to be deposited.																
5	Deposited : The transaction has been sent to be deposited.																
6	Voided: The transaction has been voided.																
7	Error: The transaction has been rejected due to an error																
8	Declined: The transaction has been declined.																
9	Verified: The Transaction has been verified.																
authorizationCode	Authorization code returned by the payment network for this transaction.																
addressVerificationResults	<table> <tr> <th>AVS Response</th><th>Text String</th></tr> </table>	AVS Response	Text String														
AVS Response	Text String																

	N1	No address given with order
	N2	Bill-to address did not pass
	““	AVS not performed (Blanks returned)
	IU	AVS not performed by Issuer
	ID	Issuer does not participate in AVS
	IE	Edit Error – AVS data is invalid
	IS	System unavailable or time-out
	IB	Street address match. Postal code not verified due to incompatible formats (both were sent).
	IC	Street address and postal code not verified due to incompatible format (both were sent).
	IP	Postal code match. Street address not verified due to incompatible formats (both were sent).
	A1	Accountholder name matches
	A3	Accountholder name, billing address and postal code.
	A4	Accountholder name and billing postal code match
	A7	Accountholder name and billing address match
	B3	Accountholder name incorrect, billing address and postal code match
	B4	Accountholder name incorrect, billing postal code matches
	B7	Accountholder name incorrect,

		billing address matches
	B8	Accountholder name, billing address and postal code are all incorrect
	??	A double question mark symbol “??” indicates an unrecognized response from association
	I1	Zip code + 4 and Address Match
	I2	Zip code +4 Match
	I3	Zip code and Address Match
	I4	Zip code Match
	I5	+4 and Address Match
	I6	+4 Match
	I7	Address Match
	I8	No Match
fraudSecurityResult	Code	Text String
	N	No Match.
	M	Fraud Security Match.
	P	Fraud Security Value not processed.
	S	No value available to check.
	T	Value on card.
	U	Fraud security value not available for this merchant.
	I	Invalid Fraud Security Value.
	<blank>	Unrecognized or not passed.
authorizedTime	Time the transaction was authorized. Uninitialized if the transaction was not successful.	
capturedTime	Time the transaction was captured. Uninitialized if the transaction was not successful.	
settledTime	Time the transaction was posted for settlement. Not the same as the time the deposit is posted in the merchant’s account.	

type	Type	Card Transaction Types
	1	Authorize
	2	Sale
	3	Credit
	4	Void
	10	Offline Force (Sale Pre-authorized)
	12	Verify

User information is returned as part of the `getPaywaySession` request.

Name	Description
name	Username of the PaywayWS user.
firstName	PaywayWS user's first name
lastName	PaywayWS user's last name
email	Email of the PaywayWS user that can be used for two-factor authentication
phone	Phone number of the PaywayWS user that can be used for two-factor authentication
timeZone	Used to synchronize PaywayWS server times to the user's time zone.

XII. Payway Result Codes (paywayCode field)

Description	paywayCode	paywayMessage
PAYWAY_WS_SUCCESS	5000	Success
PAYWAY_WS_USER_NOT_FOUND	5001	The user was not found
PAYWAY_WS_PASSWORD_EXPIRED	5002	The user's password is expired
PAYWAY_WS_INVALID_PASSWORD	5003	The user's password is invalid
PAYWAY_WS_PROHIBITED	5004	The user does not have permission to execute the request
PAYWAY_WS_NOT_LOGGED_ON	5005	The user is not logged on
PAYWAY_WS_USER_LOCKED_OUT	5007	Multiple log-on attempts have locked out this user for 30 minutes
PAYWAY_WS_USER_NOT_ACTIVATED	5008	New user whose account has not been activated

Description	paywayCode	paywayMessage
PAYWAY_WS_INVALID_USER	5009	User has been marked invalid
PAYWAY_WS_AMOUNT_EXCEEDS_USER_CREDIT_LIMIT	5010	The user has a credit limit that is less than the amount of this credit
PAYWAY_WS_CASHBOX_NOT_FOUND	5011	The cashbox (merchant) for the current transaction was not found.
PAYWAY_WS_TRANSACTION_DECLINED	5012	The transaction was declined. More information may be in the processor return code.
PAYWAY_WS_PAYMENT_TYPE_NOT_ACCEPTED	5013	unknown error: 5013
PAYWAY_WS_CARD_TYPE_NOT_ACCEPTED	5014	The merchant on Payway is not configured to allow this type of card as payment.

Description	paywayCode	paywayMessage
PAYWAY_WS_REVERSE_AUTH_FAILED	5015	The transaction was successfully voided; however, the authorization could not be cancelled.
PAYWAY_WS_PROCESSOR_ERROR	5017	An unexpected error from the processor was received. See the processor return code for more information.
PAYWAY_WS_NO_PROCESSOR_CONNECTION	5018	Payway cannot connect to the processor.
PAYWAY_WS_SOFT_DESCRIPTOR_NOT_AUTHORIZED	5019	The merchant account is not configured to send soft descriptors for payments.

Description	paywayCode	paywayMessage
PAYWAY_WS_DUPLICATE_TRANSACTION_NAME	5020	This transaction name already exists for the specified transaction source.
PAYWAY_WS_SOURCE_NOT_FOUND	5022	The specified source configuration is not found on Payway.
PAYWAY_WS_COMPANY_NOT_FOUND	5023	The company configuration on Payway is missing.
PAYWAY_WS_DIVISION_NOT_FOUND	5024	The division configuration on Payway is missing.
PAYWAY_WS_TRANSACTION_NOT_FOUND	5025	The specified transaction was not found.
PAYWAY_WS_IN_PROGRESS_TRANSACTION	5026	This transaction has already been submitted and is being processed.

Description	paywayCode	paywayMessage
PAYWAY_WS_TRANSITION_ERROR	5027	An invalid operation on the transaction was attempted that implied an invalid state transition. The transaction is in the wrong state.
PAYWAY_WS_TOKEN_NOT_ALLOWED	5028	An account token is not allowed based in the account input type specified.
PAYWAY_WS_ACCOUNT_NOT_FOUND	5029	The specified account was not found.
PAYWAY_WS_ACCOUNT_NOT_ACTIVE	5030	This account was deactivated by an administrator or by the account updater.
PAYWAY_WS_ACCOUNT_CLOSED	5031	This account was marked closed by the account updater.

Description	paywayCode	paywayMessage
PAYWAY_WS_INVALID_TRANSACTION_TYPE	5032	The transaction type specified was invalid.
PAYWAY_WS_INVALID_TRANSACTION_STATE	5033	The transaction state was invalid.
PAYWAY_WS_INVALID_TRANSACTION	5034	The transaction data is invalid.
PAYWAY_WS_INVALID_ACCOUNT_NUMBER	5035	The account number failed to pass the LUHN algorithm and is invalid.
PAYWAY_WS_INVALID_ADDRESS	5036	The address is invalid.
PAYWAY_WS_INVALID_EXPIRATION_DATE	5037	The expiration date is invalid.
PAYWAY_WS_INVALID_FRAUD_SECURITY_CODE	5038	The fraud security code is invalid.
PAYWAY_WS_INVALID_CARD_TYPE	5039	The card type is not one of the valid types.
PAYWAY_WS_INVALID_CITY	5040	The city name is invalid.
PAYWAY_WS_INVALID_FIRST_NAME	5041	The first name is invalid.

Description	paywayCode	paywayMessage
PAYWAY_WS_INVALID_LAST_NAME	5042	The last name is invalid.
PAYWAY_WS_INVALID_MIDDLE_NAME	5043	The middle name is invalid.
PAYWAY_WS_INVALID_STATE	5044	The state or province code is invalid.
PAYWAY_WS_INVALID_ZIP	5045	The zip code or postal code is invalid.
PAYWAY_WS_INVALID_PHONE	5046	The phone number is invalid.
PAYWAY_WS_INVALID_EMAIL	5047	The email is invalid.
PAYWAY_WS_INVALID_TOKEN	5048	The account token is invalid.
PAYWAY_WS_INVALID_AMOUNT	5049	The amount is invalid.
PAYWAY_WS_INVALID_PAYMENT_TYPE	5050	unknown error: 5050
PAYWAY_WS_INVALID_SALES_TAX	5051	The sales tax amount is invalid.

Description	paywayCode	paywayMessage
PAYWAY_WS_INVALID_TRANSACTION_NAME	5052	The transaction name specified is invalid.
PAYWAY_WS_INVALID_ACCOUNT_NOTES_1	5053	The data in account notes 1 field is invalid.
PAYWAY_WS_INVALID_ACCOUNT_NOTES_2	5054	The data in account notes 2 field is invalid.
PAYWAY_WS_INVALID_ACCOUNT_NOTES_3	5055	The data in account notes 3 field is invalid.
PAYWAY_WS_INVALID_ECI_TYPE	5056	The ECI type indicator is not one of the allowed choices.
PAYWAY_WS_INVALID_STATUS	5057	The transaction status is not valid when attempting an update.
PAYWAY_WS_INVALID_ACCOUNT_TYPE	5060	The account type is not one of the allowed values.

Description	paywayCode	paywayMessage
PAYWAY_WS_INVALID_TRANSACTION_NOTES_1	5061	The data in the transaction notes 1 field is not valid.
PAYWAY_WS_INVALID_TRANSACTION_NOTES_2	5062	The data in the transaction notes 1 field is not valid.
PAYWAY_WS_INVALID_TRANSACTION_NOTES_3	5063	The data in the transaction notes 1 field is not valid.
PAYWAY_WS_INVALID_AUTH_CODE	5064	The auth code is invalid for the external authorization request.
PAYWAY_WS_INVALID_TRANSACTION_ID	5065	The transaction id specified is invalid.
PAYWAY_WS_INVALID_ONLINE_PAYMENT_CRYPTOGAM	5066	The online payment cryptogram for 3D secure account input is invalid.
PAYWAY_WS_INVALID_REQUEST	5067	Invalid request type specified.

Description	paywayCode	paywayMessage
PAYWAY_WS_INVALID_INPUT_MODE	5068	Invalid input mode was specified.
PAYWAY_WS_INVALID_REQUEST_TYPE	5069	Invalid request type was specified.
PAYWAY_WS_INVALID_REQUEST_ACCOUNT_MISSING	5071	Invalid request: account information is missing.
PAYWAY_WS_INVALID_REQUEST_TRANSACTION_MISSING	5072	Invalid request: transaction information is missing.
PAYWAY_WS_TRANSACTION_TOKEN_MISSING	5073	The required transaction token was not specified.
PAYWAY_WS_TRANSACTION_QUEUED_BY_OTHER_SESSION	5074	Transaction queued by other session.
PAYWAY_WS_JSON_EXCEPTION	5076	An exception was encountered when processing the JSON payload.

Description	paywayCode	paywayMessage
PAYWAY_WS_APPLE_PAY_SERVER_IO_EXCEPTION	5077	Apple Communication Error
PAYWAY_WS_APPLE_PAY_DECRYPTION_ERROR	5078	The Apple Pay data could not be decrypted.
PAYWAY_WS_APPLE_PAY_INVALID_URL	5079	The Apple Pay merchant validation URL specified is invalid.
PAYWAY_WS_INVALID_JSON	5080	The payload is not a valid JSON object.
PAYWAY_WS_INVALID_MEDIA_TYPE	5081	The media type specified is not valid.
PAYWAY_WS_INVALID_ROUTING_NUMBER	5082	Invalid bank routing number.
PAYWAY_WS_INVALID_REPLY	5089	The reply back from the credit card processor was invalid.
PAYWAY_WS_INTERNAL_ERROR	6000	Internal Payway error. Contact Payway for assistance.

XIII. Processor Specific Result Codes

****Warning: These result codes are dependent on the customer's merchant configuration and are subject to change.***

A. TSYS

Processor Code	Text String	Comment
00	Approval	Approved and completed.
01	Call	Refer to issuer.
02	Call	Refer to issuer—Special condition
03	Term ID Error	Invalid Merchant ID
04	Hold-call	Pick up card (no fraud)
05	Decline	Do not honor
06	Error XXXX	General Error.
07	Hold-call	Pick up card, special conditions (Fraud account)
10	Partial Approval	Partial approval for authorized amount.
12	Invalid Transaction	
13	Amount Error	Invalid Amount.
14	Card No. Error	Invalid card number.
15	No such issuer	No such issuer of the credit card.
19	Re ENTER	Re-enter transaction.
39	No Credit Acct	No Credit Account.
41	Hold-call	Lost card, pick up card (fraud account).
43	Hold-call	Pick up card, stolen card, pick up (fraud account).
51	Decline	Insufficient funds.

Processor Code	Text String	Comment
54	Expired card	Expired card—different than invalid expiration date.
57	Service not allowed	Transaction not permitted on card.
58	Service not allowed	Service not allowed. Transaction not permitted—Terminal.
59	Service not allowed	Service not allowed. Transaction not permitted—Merchant.
61	Declined	Exceeds withdrawal limit.
62	Declined	Invalid service code, restricted.
63	Sec Violation	Security violation.
65	Declined	Activity limit exceeded.
77	No Action taken	Inconsistent data, reversed or repeat.
79	Already Reversed	Already reversed at switch.
80	No impact	No financial impact (used in responses for declined originals).
82	Incorrect CVV	CVV data not correct.
85	Card OK	No reason to decline.
93	Decline violation	Cannot complete.
94	Duplicate Trans	Unable to locate, no match.
96	System Error	System malfunction.
CV	Failure CV	Card Type verification error.

Processor Code	Text String	Comment
EC	CID format error	Verification error.
HV	Failure Hierarchy	Verification error.
N4	Decline	Exceeds Issuer withdrawal limit.
N7	CCV2 mismatch	CVV2 value supplied is invalid.
R0	Stop recurring	Customer requested stop of specific recurring transactions.

B. First Data Merchant Services and Chase Paymentech

Processor Code	Text String	Comment
100	Approved	Successfully approved.
201	Invalid Account Number	Bad check digit, length, or other problem.
202	Bad amount	Amount sent was zero.
203	Bad amount	Amount sent was zero.
204	Other Error	Unidentifiable error.
205	Bad total amount	Amount sent was zero.
223	Does not match MOP	Credit card number does not match method of payment. (MOP)
253	Invalid tran type	Invalid transaction type for this order.
302	Credit floor	Insufficient funds.
303	Processor decline	Generic decline, no other reason given provided by the card issuer.
304	Not on file	Non-existent account.
305	Already reversed	Transaction previously reversed.
401	Decline	Decline.
402	Default Call	Decline.
501	Pickup	Card stolen.
502	Lost/Stolen	Card reported as lost or stolen.
503	Security violation	Fraud security code does not match.
505	Negative file	Account on negative file list.
509	Over the limit	Exceed withdrawal or activity limit.
510	Over limit frequency count	Exceed withdrawal or activity count limit.
521	Insufficient funds	Insufficient funds or over credit limit.
522	Card is expired	Card has expired—not the same as invalid expiry date.
530	Do not honor	Generic decline—no reason given by card issuer.
531	CVV2 failure	Issuer declined because CVV2 failed.
550	Closed account	New account issued.

Processor Code	Text String	Comment
570	Stop payment	Cardholder requests recurring payments be stopped.
571	Revocation	Cardholder requests recurring payments be stopped.
591	Bad account number	Bad check digit or other problem.
594	Other error	Unidentifiable error issuer generated.
596	Suspected Fraud	Issuer has flagged this account as fraudulent.
605	Bad expiration date	Card has expired.
606	Invalid Transaction type	Issuer does not allow this type of transaction.
806	Restraint	Card has been restricted.
825	No account	Account does not exist.

XIV. Request / Reply Examples

A. Get A Payway Session

path: /Session
method: POST

request:

```
{  
  "request": "getPaywaySession",  
  "companyId": 22,  
  "password": "1234567abc",  
  "userName": "villagePerson"  
}
```

getPaywaySession Reply

```
{  
  "paywayCode": "5000",  
  "paywayMessage": "",  
  "paywaySessionToken":  
  "85EAE7B823DFEA917E61610580794A97",  
  "user": {  
    "email": "",  
    "firstName": "",  
    "lastName": "",  
    "name": "villagePerson",  
    "notes": "",  
    "phone": "",  
    "timeZone": "America/New_York"  
  }  
}
```

B. Trial Authorize with Primary Account Number

```
path: /Payment/CreditCard method: POST
{
  "accountInputMode": "primaryAccountNumber",
  "cardAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "4012888888881881",
    "address": "12 Red Rock Terrace",
    "city": "Singing Desert",
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "07/24",
    "firstName": "Wiley",
    "fsv": "",
    "lastName": "Coyote",
    "middleName": "Genius",
    "phone": "222-222-2222",
    "state": "AZ",
    "zip": "01929"
  },
  "cardTransaction": {
    "name": "20210604_01_CC_TrialWithAccount",
    "sourceId": "2",
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note",
  },
  "paywaySessionToken": "C9AFCB004E38DFC08AFF651E4A51264C",
  "request": "trialAuthorize"
}
```


Reply

```
{
  "cardAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "401288*****1881",
    "account_number_masked": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "auLastUpdate": "1999-01-01 00:00",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "divisionId": 1,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "0724",
    "firstFour": "4012",
    "firstName": "Wiley",
    "fsv": "",
    "inputMode": 1,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "Genius",
    "onlinePaymentCryptogram": "",
    "p2peInput": "",
    "paywayToken": 10000129,
    "phone": "2222222222",
    "state": "AZ",
    "status": 0,
    "zip": "01929"
  },
  "cardTransaction": {
    "addressVerificationResults": "I8",
    "amount": 0,
    "authorizationCode": "408162",
```

```
"authorizedTime": "2021-06-04",
"capturedTime": "1999-01-01 00:00:00-05",
"cbMode": 4,
"eciType": 1,
"fraudSecurityResults": " ",
"fsvIndicator": "",
"name": "20210604_01_CC_TrialWithAccount",
"pfpstatus": 3601,
"pfpstatusString": "PFP Not Enabled",
"processorErrorMessage": "",
"processorOrderId": "",
"processorRecurringAdvice": " ",
"processorResponseDate": "210604",
"processorResultCode": "100",
"processorSequenceNumber": 0,
"processorSoftDescriptor": "",
"referenceNumber": "",
"resultCode": 0,
"settledTime": "1999-01-01 00:00",
"sourceId": 2,
"status": 9,
"tax": 0,
"testResultAVS": "",
"testResultFSV": "",
"transactionNotes1": "Notes field 1",
"transactionNotes2": "Order 123",
"transactionNotes3": "Other note",
"type": 12
},
"paywayCode": "5000",
"paywayMessage": ""
}
```

C. Authorize with Primary Account Number

```
path: /Payment/CreditCard
method: POST
Request:{
  "accountInputMode": "primaryAccountNumber",
  "cardAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "4012888888881881",
    "address": "12 Red Rock Terrace",
    "city": "Singing Desert",
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "11/24",
    "firstName": "Wiley",
    "fsv": "",
    "lastName": "Coyote",
    "middleName": "Genius",
    "phone": "222-222-2222",
    "state": "AZ",
    "zip": "01929"
  },
  "cardTransaction": {
    "amount": 1000,
    "eciType": 1,
    "name": "TEST-CC-117-03",
    "processorSoftDescriptor": "Joe's Taxi 45th St",
    "sourceId": 11,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
  "paywaySessionToken": "3BBE7C8F1FB42C5BC9E9FD4A41CE1983",
  "request": "authorize"
}
```

```
Reply:{
  "cardAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "401288*****1881",
    "account_number_masked": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "divisionId": 7,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "1124",
    "firstFour": "4012",
    "firstName": "Wiley",
    "fsv": "",
    "inputMode": 1,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": "1999-01-01 00:00:00-05",
    "middleName": "Genius",
    "onlinePaymentCryptogram": "",
    "p2peInput": "",
    "paywayToken": 10158444,
    "phone": "2222222222",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  },
  "cardTransaction": {
    "addressVerificationResults": "",
    "amount": 1000,
    "authorizationCode": "0987654321",
    "authorizedTime": "2021-02-25 00:00:00-05",
```

```
"capturedTime": "2021-02-25 08:47:03",
"cbMode": 2,
"eciType": 1,
"fraudSecurityResults": "",
"fsvIndicator": "",
"name": "TEST-CC-117-03",
"pfpstatus": 3601,
"pfpstatusString": "PFP Not Enabled",
"processorErrorMessage": "",
"processorOrderId": "",
"processorRecurringAdvice": "",
"processorResponseDate": "",
"processorResultCode": "",
"processorSequenceNumber": 0,
"processorSoftDescriptor": "Joe's Taxi 45th St",
"referenceNumber": "123456",
"resultCode": 0,
"sessionToken_string": "0",
"settledTime": "1999-01-01 00:00",
"sourceId": 11,
"status": 3,
"tax": 100,
"testResultAVS": "",
"testResultFSV": "",
"transactionNotes1": "Notes field 1",
"transactionNotes2": "Order 123",
"transactionNotes3": "Other note",
"type": 1,
},
"paywayCode": "5000",
"paywayMessage": ""
}
```

D. Sale with Primary Account Number

```
path: /Payment/CreditCard method: POST
{
  "accountInputMode": "primaryAccountNumber",
  "cardAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "4012888888881881",
    "address": "12 Red Rock Terrace",
    "city": "Singing Desert",
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "07/19",
    "firstName": "Wiley",
    "fsv": "",
    "lastName": "Coyote",
    "middleName": "Genius",
    "phone": "222-222-2222",
    "state": "AZ",
    "zip": "01929"
```

```
},
"cardTransaction": {
  "amount": 1000,
  "eciType": 1,
  "sourceId": 20,
  "name": "TEST-81-00",
  "processorSoftDescriptor": "Joe's Taxi 45th St",
  "tax": 100,
  "transactionNotes1": "Notes field 1",
  "transactionNotes2": "Order 123",
  "transactionNotes3": "Other note"
},
"paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",
"request": "sale"
}
```

sale Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": "",

  "cardAccount": {
    "accountNotes1": "",
    "accountNotes2": "",
    "accountNotes3": "Info about account",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "0719",
    "firstFour": "4012",
    "firstName": "Wiley",
    "fsv": "",
    "divisionId": 17,
    "inputMode": 4,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": null,
    "middleName": "Genius",
    "p2peInput": "",
    "paywayAccountToken": 10117024,
    "phone": "2222222222",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  },
  "cardTransaction": {
    "addressVerificationResults": "",
    "amount": 1000,
```



```
"authorizationCode": "0987654321",
"authorizedTime": "2020-04-30 00:00:00-04",
"capturedTime": "2020-04-30 16:41:29",
"cbMode": 2,
"eciType": 1,
"fraudSecurityResults": "",
"fsvIndicator": "",
"sourceId": 20,
"name": "TEST-81-00",
"pfpstatus": 1,
"processorErrorMessage": "",
"processorOrderId": "",
"processorRecurringAdvice": "",
"processorResponseDate": "",
"processorResultCode": "",
"processorSequenceNumber": 0,
"processorSoftDescriptor": "Joe's Taxi 45th St",
"referenceNumber": "123456",
"resultCode": 0,
"sessionToken_string": "485",
"settledTime": "1999-01-01 00:00",
"status": 4,
"tax": 100,
"testResultAVS": "",
"testResultFSV": "",
"transactionNotes1": "Notes field 1",
"transactionNotes2": "Order 123",
"transactionNotes3": "Other note",
"type": 2,
},
}
```

E. Pre Authorized Sale with Primary Account Number

```
path : "/Payment/CreditCard" method : POST
{
  "request" : "salePreAuthorized",
  "accountInputMode" : "primaryAccountNumber",
  "authCode" : "123123",
  "cardTransaction" :
  {
    "eciType" : 5,
    "name" : "TEST-104-09",
    "sourceld" : 20,
    "amount" : 1000,
    "tax" : 100,
    "processorSoftDescriptor" : "Joe's Taxi 45th St",
    "transactionNotes1" : "Notes field 1",
    "transactionNotes2" : "Order 123",
    "transactionNotes3" : "Other note"
  },
  "cardAccount" :
  {
    "fsv" : "",
    "accountNumber" : "4012888888881881",
    "expirationDate" : "07/19",
    "email" : "dan.r.Coyote@gmail.com",
    "firstName" : "Wiley",
    "lastName" : "Coyote",
    "middleName" : "Genius",
    "address" : "12 Red Rock Terrace",
    "city" : "Singing Desert",
    "state" : "AZ",
    "zip" : "01929",
    "phone" : "222-222-2222",
    "accountNotes1" : "Notes about account",
    "accountNotes2" : "More info",
    "accountNotes3" : ""
  }
}
```

```
}  
Reply:  
{  
  "paywayCode": "5000",  
  "paywayMessage": "",  
  
  "cardAccount": {  
    "accountNotes1": "",  
    "accountNotes2": "",  
    "accountNotes3": "Info about account",  
    "accountNumber": "401288*****1881",  
    "address": "12 Red Rock Terrace",  
    "addressList": null,  
    "auLastUpdate": "1999-01-01 00:00:00-05",  
    "auUpdateType": 0,  
    "cardType": 1,  
    "city": "Singing Desert",  
    "commercialCardType": 0,  
    "divisionId": 17,  
    "email": "dan.r.Coyote@gmail.com",  
    "expirationDate": "0719",  
    "firstFour": "4012",  
    "firstName": "Wiley",  
    "fsv": "",  
    "inputMode": 4,  
    "lastFour": "1881",  
    "lastName": "Coyote",  
    "lastUsed": null,  
    "middleName": "Genius",  
    "onlinePaymentCryptogram": "",  
    "p2peInput": "",  
    "paywayAccountToken": 10117024,  
    "phone": "2222222222",  
    "state": "AZ",  
    "status": 2,  
    "zip": "01929"  
  },  
  "cardTransaction":
```

```
{
  "addressVerificationResults": "",
  "amount": 1000,
  "authorizationCode": "",
  "authorizedTime": "1999-01-01 00:00",
  "capturedTime": "1999-01-01 00:00",
  "cbMode": 2,
  "eciType": 2,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "name": "TEST-104-10",
  "pfpstatus": 0,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "Acme Monthly Subscription",
  "referenceNumber": "",
  "resultCode": 0,
  "sessionToken_string": "0",
  "settledTime": "1999-01-01 00:00",
  "sourceId": 20,
  "status": 2,
  "tax": 100,
  "testResultAVS": "",
  "testResultFSV": "",
  "transactionNotes1": "",
  "transactionNotes2": "",
  "transactionNotes3": ""
},
```

F. Credit With Primary Account Number

```
path: /Payment/CreditCard method: POST
{
  "accountInputMode": "primaryAccountNumber",
  "cardAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "4012888888881881",
    "address": "12 Red Rock Terrace",
    "city": "Singing Desert",
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "07/19",
    "firstName": "Wiley",
    "fsv": "",
    "lastName": "Coyote",
    "middleName": "Genius",
    "phone": "222-222-2222",
    "state": "AZ",
    "zip": "01929"
  },
  "cardTransaction": {
    "amount": 1000,
    "eciType": 1,
    "sourceId": 20,
    "name": "TEST-81-01",
    "processorSoftDescriptor": "Joe's Taxi 45th St",
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",
  "request": "credit"
}
```

credit Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": ""

  "cardAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "0719",
    "firstFour": "4012",
    "firstName": "Wiley",
    "fsv": "",
    "divisionId": 17,
    "inputMode": 4,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "Genius",
    "p2peInput": "",
    "paywayAccountToken": 10117024,
    "phone": "2222222222",
    "state": "AZ",
    "status": 0,
    "zip": "01929"
  },
  "cardTransaction": {
    "addressVerificationResults": "",
```

```
"amount": 1000,
"authorizationCode": "",
"authorizedTime": "2020-04-30 16:41:31",
"capturedTime": "2020-04-30 16:41:31",
"cbMode": 2,
"eciType": 1,
"fraudSecurityResults": "",
"fsvIndicator": "",
"sourceId": 20,
"name": "TEST-81-01",
"pfpsstatus": 0,
"processorErrorMessage": "",
"processorOrderId": "",
"processorRecurringAdvice": "",
"processorResponseDate": "",
"processorResultCode": "",
"processorSequenceNumber": 0,
"processorSoftDescriptor": "Joe's Taxi 45th St",
"referenceNumber": "",
"resultCode": 0,
"sessionToken_string": "0",
"settledTime": "1999-01-01 00:00",
"status": 4,
"tax": 100,
"testResultAVS": "",
"testResultFSV": "",
"transactionNotes1": "Notes field 1",
"transactionNotes2": "Order 123",
"transactionNotes3": "Other note",
"type": 3,
}
}
```

G. Sale with Payway Account Token

path: /Payment/CreditCard method: POST

```
{
  "request": "sale"
  "accountInputMode": "paywayAccountToken",
  "paywayAccountToken": "10117024",
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",

  "cardTransaction": {
    "amount": 1000,
    "eciType": 1,
    "sourceId": 20,
    "name": "TEST-81-03",
    "processorSoftDescriptor": "Acme Monthly Subscription",
    "tax": 100,
    "transactionNotes1": "",
    "transactionNotes2": "",
    "transactionNotes3": ""
  },
}
```

sale Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": ""

  "cardAccount": {
    "accountNotes1": "",
    "accountNotes2": "",
    "accountNotes3": "Info about account",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00:00-05",
  }
}
```



```
"auUpdateType": 0,
"cardType": 1,
"city": "Singing Desert",
"commercialCardType": 0,
"email": "dan.r.Coyote@gmail.com",
"expirationDate": "0719",
"firstFour": "4012",
"firstName": "Wiley",
"fsv": "",
"divisionId": 17,
"inputMode": 4,
"lastFour": "1881",
"lastName": "Coyote",
"lastUsed": null,
"middleName": "Genius",
"p2peInput": "",
"paywayAccountToken": 10117024,
"phone": "2222222222",
"state": "AZ",
"status": 2,
"zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 1000,
  "authorizationCode": "0987654321",
  "authorizedTime": "2020-04-30 00:00:00-04",
  "capturedTime": "2020-04-30 16:41:34",
  "cbMode": 2,
  "eciType": 2,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "sourceId": 20,
  "name": "TEST-81-03",
  "pfpstatus": 1,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
```

```
"processorResponseDate": "",
"processorResultCode": "",
"processorSequenceNumber": 0,
"processorSoftDescriptor": "Acme Monthly Subscription",
"referenceNumber": "123456",
"resultCode": 0,
"sessionToken_string": "487",
"settledTime": "1999-01-01 00:00",
"status": 4,
"tax": 100,
"testResultAVS": "",
"testResultFSV": "",
"transactionNotes1": "",
"transactionNotes2": "",
"transactionNotes3": "",
"type": 2,

}

}
```

H. Trial Authorize with Payway Account Token

path: /Payment/CreditCard method: POST

```
{
  "cardTransaction": {
    "name": "20210604_01_CC_TrialWithToken",
    "sourceId": "2",
    "transactionNotes1": "",
    "transactionNotes2": "",
    "transactionNotes3": ""
  },
  "paywayAccountToken": "010000130",
  "paywaySessionToken": "C9AFCB004E38DFC08AFF651E4A51264C",
  "request": "trialAuthorize"
}
```

Reply

```
{
  "cardAccount": {
    "accountNotes1": "UPDATED_Here",
    "accountNotes2": "UPDATED_is my",
    "accountNotes3": "UPDATED_account",
    "accountNumber": "401288*****1881",
    "account_number_masked": "401288*****1881",
    "address": "UPDATED_12 Red Rock Terrace",
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "UPDATED_Singing Desert",
    "commercialCardType": 0,
    "divisionId": 1,
    "email": "UPDATED_dan.r.Coyote@gmail.com",
    "expirationDate": "0725",
    "firstFour": "4012",
    "firstName": "UPDATED_Wiley",
    "fsv": "",
    "inputMode": 1,
    "lastFour": "1881",
  }
}
```

```
"lastName": "UPDATED_Coyote",
"lastUsed": "1999-01-01 00:00:00-05",
"middleName": "UPDATED_Genius",
"onlinePaymentCryptogram": "",
"p2peInput": "",
"paywayToken": 10000130,
"phone": "3333333333",
"state": "MA",
"status": 2,
"zip": "88888"
},
"cardTransaction": {
  "addressVerificationResults": "I8",
  "amount": 0,
  "authorizationCode": "541580",
  "authorizedTime": "2021-06-04",
  "capturedTime": "1999-01-01 00:00:00-05",
  "cbMode": 4,
  "eciType": 1,
  "fraudSecurityResults": " ",
  "fsvIndicator": "",
  "name": "20210604_01_CC_TrialWithToken",
  "pfpstatus": 3601,
  "pfpstatusString": "PFP Not Enabled",
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": " ",
  "processorResponseDate": "210604",
  "processorResultCode": "100",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "",
  "referenceNumber": "",
  "resultCode": 0,
  "settledTime": "1999-01-01 00:00",
  "sourceId": 2,
  "status": 9,
  "tax": 0,
  "testResultAVS": "",
```

```
"testResultFSV": "",
"transactionNotes1": "",
"transactionNotes2": "",
"transactionNotes3": "",
"type": 12
},
"paywayCode": "5000",
"paywayMessage": ""
}
```

I. Pre Authorized Sale with Payway Account Token

path: /Payment/CreditCard

method: POST

```
Request:{
  "paywayAccountToken": "10117024",
  "paywaySessionToken": "9A52A3F9E85FF0605DA588BAA9894D79",
  "request": "salePreAuthorized"
  "authCode": "123123123",

  "cardTransaction": {
    "amount": 1000,
    "eciType": 1,
    "name": "TEST-104-10",
    "processorSoftDescriptor": "Acme Monthly Subscription",
    "sourceId": 20,
    "tax": 100,
    "transactionNotes1": "",
    "transactionNotes2": "",
    "transactionNotes3": ""
  }
}
```

sale Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

```
"cardAccount": {
  "accountNotes1": "",
  "accountNotes2": "",
  "accountNotes3": "Info about account",
  "accountNumber": "401288*****1881",
  "address": "12 Red Rock Terrace",
  "addressList": null,
  "auLastUpdate": "1999-01-01 00:00:00-05",
  "auUpdateType": 0,
  "cardType": 1,
  "city": "Singing Desert",
  "commercialCardType": 0,
  "email": "dan.r.Coyote@gmail.com",
  "expirationDate": "0719",
  "firstFour": "4012",
  "firstName": "Wiley",
  "fsv": "",
  "divisionId": 17,
  "inputMode": 4,
  "lastFour": "1881",
  "lastName": "Coyote",
  "lastUsed": null,
  "middleName": "Genius",
  "p2peInput": "",
  "paywayAccountToken": 10117024,
  "phone": "2222222222",
  "state": "AZ",
  "status": 2,
  "zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 1000,
  "authorizationCode": "0987654321",
  "authorizedTime": "2020-04-30 00:00:00-04",
  "capturedTime": "2020-04-30 16:41:34",
  "cbMode": 2,
  "eciType": 2,
```

```
"fraudSecurityResults": "",
"fsvIndicator": "",
"sourceId": 20,
"name": "TEST-81-03",
"pfpstatus": 1,
"processorErrorMessage": "",
"processorOrderId": "",
"processorRecurringAdvice": "",
"processorResponseDate": "",
"processorResultCode": "",
"processorSequenceNumber": 0,
"processorSoftDescriptor": "Acme Monthly Subscription",
"referenceNumber": "123456",
"resultCode": 0,
"sessionToken_string": "487",
"settledTime": "1999-01-01 00:00",
"status": 4,
"tax": 100,
"testResultAVS": "",
"testResultFSV": "",
"transactionNotes1": "",
"transactionNotes2": "",
"transactionNotes3": "",
"type": 2,
}
}
```

J. Credit with Payway Account Token

path: /Payment/CreditCard method: POST

```
{
  "cardTransaction": {
    "amount": 1000,
    "eciType": 1,
    "sourceId": 20,
    "name": "TEST-81-04",
    "processorSoftDescriptor": "Acme Monthly Subscription",
    "tax": 100,
    "transactionNotes1": "",
    "transactionNotes2": "",
    "transactionNotes3": ""
  },
  "paywayAccountToken": "10117024",
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",
  "request": "credit"
}
```

credit Reply

```
{

  "paywayCode": "5000",
  "paywayMessage": "",

  "cardAccount": {
    "accountNotes1": "",
    "accountNotes2": "",
    "accountNotes3": "Info about account",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
```



```
"commercialCardType": 0,  
"email": "dan.r.Coyote@gmail.com",  
"expirationDate": "0719",  
"firstFour": "4012",  
"firstName": "Wiley",  
"fsv": "",  
"divisionId": 17,  
"inputMode": 4,  
"lastFour": "1881",  
"lastName": "Coyote",  
"lastUsed": null,  
"middleName": "Genius",  
"p2peInput": "",  
"paywayAccountToken": 10117024,  
"phone": "2222222222",  
"state": "AZ",  
"status": 2,  
"zip": "01929"  
},  
"cardTransaction": {  
  "addressVerificationResults": "",  
  "amount": 1000,  
  "authorizationCode": "",  
  "authorizedTime": "2020-04-30 16:41:35",  
  "capturedTime": "2020-04-30 16:41:35",  
  "cbMode": 2,  
  "eciType": 2,  
  "fraudSecurityResults": "",  
  "fsvIndicator": "",  
  "sourceId": 20,  
  "name": "TEST-81-04",  
  "pfpstatus": 0,  
  "processorErrorMessage": "",  
  "processorOrderId": "",  
  "processorRecurringAdvice": "",  
  "processorResponseDate": "",  
  "processorResultCode": "",  
  "processorSequenceNumber": 0,
```



```
"processorSoftDescriptor": "Acme Monthly Subscription",  
"referenceNumber": "",  
"resultCode": 0,  
"sessionToken_string": "0",  
"settledTime": "1999-01-01 00:00",  
"status": 4,  
"tax": 100,  
"testResultAVS": "",  
"testResultFSV": "",  
"transactionNotes1": "",  
"transactionNotes2": "",  
"transactionNotes3": "",  
"type": 3,  
  
}}
```

K. Authorize with Payway Account Token

path: /Payment/CreditCard method: POST

```
{
  "cardTransaction": {
    "amount": 1000,
    "eciType": 1,
    "sourceId": 20,
    "name": "TEST-81-05",
    "tax": 100,
    "transactionNotes1": "",
    "transactionNotes2": "",
    "transactionNotes3": ""
  },
  "paywayAccountToken": "10117024",
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",
  "request": "authorize"
}
```

authorize Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": "",

  "cardAccount": {
    "accountNotes1": "",
    "accountNotes2": "",
    "accountNotes3": "Info about account",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
  }
}
```

```
"email": "dan.r.Coyote@gmail.com",
"expirationDate": "0719",
"firstFour": "4012",
"firstName": "Wiley",
"fsv": "",
"divisionId": 17,
"inputMode": 4,
"lastFour": "1881",
"lastName": "Coyote",
"lastUsed": null,
"middleName": "Genius",
"p2peInput": "",
"paywayAccountToken": 10117024,
"phone": "2222222222",
"state": "AZ",
"status": 2,
"zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 1000,
  "authorizationCode": "0987654321",
  "authorizedTime": "2020-04-30",
  "capturedTime": "1999-01-01 00:00:00-05",
  "cbMode": 2,
  "eciType": 1,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "sourceId": 20,
  "name": "TEST-81-05",
  "pfpstatus": 1,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "",
```

```
"referenceNumber": "123456",
"resultCode": 0,
"sessionToken_string": "0",
"settledTime": "1999-01-01 00:00",
"status": 3,
"tax": 100,
"testResultAVS": "",
"testResultFSV": "",
"transactionNotes1": "",
"transactionNotes2": "",
"transactionNotes3": "",
"type": 1,

}

}
```



L. Sale with P2PE Data

[illegible]



```
"request": "sale"
}
```

[illegible]

```
"state": "AZ",
"status": 0,
"zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 0,
  "authorizationCode": "",
  "authorizedTime": "1999-01-01",
  "capturedTime": "1999-01-01",
  "cbMode": 0,
  "eciType": 0,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "sourceId": 0,
  "name": "",
  "pfpstatus": 0,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "",
  "referenceNumber": "",
  "resultCode": 1,
  "sessionToken_string": "0",
  "settledTime": "1999-01-01 00:00",
  "status": 0,
  "tax": 0,
  "testResultAVS": "",
  "testResultFSV": "",
  "transactionNotes1": "",
  "transactionNotes2": "",
  "transactionNotes3": "",
  "type": 2,
}
```


}



M. Credit with P2PE Data

[illegible]



```
"request": "credit"
}
```

[illegible]

```
"state": "AZ",
"status": 0,
"zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 0,
  "authorizationCode": "",
  "authorizedTime": "1999-01-01",
  "capturedTime": "1999-01-01",
  "cbMode": 0,
  "eciType": 0,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "sourceId": 0,
  "name": "",
  "pfpstatus": 0,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "",
  "referenceNumber": "",
  "resultCode": 1,
  "sessionToken_string": "0",
  "settledTime": "1999-01-01 00:00",
  "status": 0,
  "tax": 0,
  "testResultAVS": "",
  "testResultFSV": "",
  "transactionNotes1": "",
  "transactionNotes2": "",
  "transactionNotes3": "",
  "type": 3,
},
```

}



N. Authorize with P2PE Data

[illegible]



```
"request": "authorize"
}
```

```
authorize Reply
{
    "paywayCode": "5000",
    "paywayMessage": ""

    "cardAccount": {
        "accountNotes1": "Notes about account",
        "accountNotes2": "More info",
        "accountNotes3": "",
        "accountNumber": "",
        "address": "12 Red Rock Terrace",
        "addressList": null,
        "auLastUpdate": "1999-01-01 00:00",
        "auUpdateTime": 0,
        "cardType": 8,
        "city": "Singing Desert",
        "commercialCardType": 0,
        "email": "dan.r.Coyote@gmail.com",
        "expirationDate": "",
        "firstFour": "",
        "firstName": "Wiley",
        "fsv": "",
        "divisionId": 17,
        "inputMode": 3,
        "lastFour": "",
        "lastName": "Coyote",
        "lastUsed": "1999-01-01 00:00",
        "middleName": "Genius",
        "p2pelnput":
"02A600C0170018008292;4012*****1881=2012?*302D7200311170547D3823
D011FA882D0B412F34D366406900000000000000000000000000000000000000
0038333755313135353533629949960E004640009E185C03",
        "paywayAccountToken": 0,
        "phone": "2222222222",
```

```
"state": "AZ",
"status": 0,
"zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 0,
  "authorizationCode": "",
  "authorizedTime": "1999-01-01",
  "capturedTime": "1999-01-01",
  "cbMode": 0,
  "eciType": 0,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "sourceId": 0,
  "name": "",
  "pfpstatus": 0,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "",
  "referenceNumber": "",
  "resultCode": 1,
  "sessionToken_string": "0",
  "settledTime": "1999-01-01 00:00",
  "status": 0,
  "tax": 0,
  "testResultAVS": "",
  "testResultFSV": "",
  "transactionNotes1": "",
  "transactionNotes2": "",
  "transactionNotes3": "",
  "type": 1,
},
```


}

O. Capture Authorized Transaction

path: /Payment/CreditCard method: POST

```
{
  "cardTransaction": {
    "sourceId": 20,
    "name": "TEST-81-05"
  },
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",
  "request": "capture"
}
```

capture Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": "",

  "cardAccount": {
    "accountNotes1": "",
    "accountNotes2": "",
    "accountNotes3": "Info about account",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "0719",
    "firstFour": "4012",
    "firstName": "Wiley",
    "fsv": "",
    "divisionId": 17,
    "inputMode": 4,
  }
}
```

```
"lastFour": "1881",
"lastName": "Coyote",
"lastUsed": null,
"middleName": "Genius",
"p2peInput": "",
"paywayAccountToken": 10117024,
"phone": "2222222222",
"state": "AZ",
"status": 2,
"zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 1000,
  "authorizationCode": "0987654321",
  "authorizedTime": "2020-04-30 00:00:00-04",
  "capturedTime": "2020-04-30 16:41:41",
  "cbMode": 2,
  "eciType": 2,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "sourceId": 20,
  "name": "TEST-81-05",
  "pfpstatus": 0,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "",
  "referenceNumber": "123456",
  "resultCode": 0,
  "sessionToken_string": "488",
  "settledTime": "1999-01-01 00:00",
  "status": 4,
  "tax": 100,
  "testResultAVS": "",
```

```
"testResultFSV": "",  
"transactionNotes1": "",  
"transactionNotes2": "",  
"transactionNotes3": "",  
"type": 2,  
  
}  
  
}
```

P. Void Transaction

path: /Payment/CreditCard method: POST

```
{
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",
  "request": "void"

  "cardTransaction": {
    "sourceld": 20,
    "name": "TEST-81-05"
  },
}
```

void Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": "",

  "cardAccount": {
    "accountNotes1": "",
    "accountNotes2": "",
    "accountNotes3": "Info about account",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "0719",
    "firstFour": "4012",
    "firstName": "Wiley",
    "fsv": "",
  },
}
```

```
"divisionId": 17,
"inputMode": 4,
"lastFour": "1881",
"lastName": "Coyote",
"lastUsed": null,
"middleName": "Genius",
"p2peInput": "",
"paywayAccountToken": 10117024,
"phone": "2222222222",
"state": "AZ",
"status": 2,
"zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 1000,
  "authorizationCode": "0987654321",
  "authorizedTime": "2020-04-30 00:00:00-04",
  "capturedTime": "2020-04-30 16:41:41-04",
  "cbMode": 2,
  "eciType": 2,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "sourceId": 20,
  "name": "TEST-81-05",
  "pfpstatus": 0,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "",
  "referenceNumber": "123456",
  "resultCode": 0,
  "sessionToken_string": "0",
  "settledTime": "1999-01-01 00:00",
  "status": 6,
```

```
"tax": 100,  
"testResultAVS": "",  
"testResultFSV": "",  
"transactionNotes1": "",  
"transactionNotes2": "",  
"transactionNotes3": "",  
"type": 4,  
}  
}
```

Q. Add New Credit Card with Primary Account Data

```
path: /Account/CreditCard method: PUT  
{  
  "paywayCode": "5000",  
  "paywayMessage": ""  
  
  "accountInputMode": "primaryAccountNumber",  
  "cardAccount": {  
    "accountNotes1": "Here",  
    "accountNotes2": "is my",  
    "accountNotes3": "account",  
    "accountNumber": "5555553753048194",  
    "address": "12 Red Rock Terrace",  
    "city": "Singing Desert",  
    "email": "dan.r.Coyote@gmail.com",  
    "expirationDate": "07/25",  
    "firstName": "Wiley",  
    "fsv": "",  
    "divisionId": "17",  
    "lastName": "Coyote",  
    "middleName": "Genius",  
    "phone": "222-222-2222",  
    "state": "AZ",  
    "zip": "01929"  
  },  
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",  
}
```

```
"request": "addAccount"
}
```

addAccount Reply

```
{
  "cardAccount": {
    "accountNotes1": "Here",
    "accountNotes2": "is my",
    "accountNotes3": "account",
    "accountNumber": "555555*****8194",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00",
    "auUpdateType": 0,
    "cardType": 2,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "0725",
    "firstFour": "5555",
    "firstName": "Wiley",
    "fsv": "",
    "divisionId": 17,
    "inputMode": 1,
    "lastFour": "8194",
    "lastName": "Coyote",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "Genius",
    "p2peInput": "",
    "paywayAccountToken": 10117053,
    "phone": "2222222222",
    "state": "AZ",
    "status": 0,
    "zip": "01929"
  }
}
```




R. Add New Account With P2PE Data

[illegible]

```
addAccount Reply
{
  "paywayCode": "5000",
  "paywayMessage": ""
```

[illegible]

S. Retrieve Account Data

path: /Query/CreditCard method: POST

```
{
  "paywayAccountToken": "10117024",
  "divisionId": "17",
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",
  "request": "getAccount"
}
```

getAccount Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": "",

  "cardAccount": {
    "accountNotes1": "",
    "accountNotes2": "",
    "accountNotes3": "Info about account",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "0719",
    "firstFour": "4012",
    "firstName": "Wiley",
    "fsv": "",
    "divisionId": 17,
    "inputMode": 4,
    "lastFour": "1881",
    "lastName": "Coyote",
  }
}
```

```
"lastUsed": null,  
"middleName": "Genius",  
"p2peInput": "",  
"paywayAccountToken": 10117024,  
"phone": "2222222222",  
"state": "AZ",  
"status": 2,  
"zip": "01929"  
}  
}
```

T. Retrieve Transaction Data

path: /Query/CreditCard method: POST

```
{  
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",  
  "request": "getTransaction",  
  "sourceId": "20",  
  "transactionName": "TEST-81-04"  
}
```

getTransaction Reply

```
{  
  "paywayCode": "5000",  
  "paywayMessage": "",  
  
  "cardAccount": {  
    "accountNotes1": "",  
    "accountNotes2": "",  
    "accountNotes3": "Info about account",  
    "accountNumber": "401288*****1881",  
    "address": "12 Red Rock Terrace",  
    "addressList": null,  
    "auLastUpdate": "1999-01-01 00:00:00-05",  
    "auUpdateType": 0,  
    "cardType": 1,  
    "city": "Singing Desert",  
    "commercialCardType": 0,  
    "email": "dan.r.Coyote@gmail.com",  
    "expirationDate": "0719",  
    "firstFour": "4012",  
    "firstName": "Wiley",  
    "fsv": "",  
    "divisionId": 17,  
    "inputMode": 4,  
    "lastFour": "1881",  
    "lastName": "Coyote",  
  }  
}
```

```
"lastUsed": null,
"middleName": "Genius",
"p2peInput": "",
"paywayAccountToken": 10117024,
"phone": "2222222222",
"state": "AZ",
"status": 2,
"zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 1000,
  "authorizationCode": "",
  "authorizedTime": "2020-04-30 16:41:35-04",
  "capturedTime": "2020-04-30 16:41:35-04",
  "cbMode": 2,
  "eciType": 2,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "sourceId": 20,
  "name": "TEST-81-04",
  "pfpstatus": 0,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "Acme Monthly Subscription",
  "referenceNumber": "",
  "resultCode": 5000,
  "sessionToken_string": "0",
  "settledTime": "1999-01-01 00:00",
  "status": 4,
  "tax": 100,
  "testResultAVS": "",
  "testResultFSV": "",
  "transactionNotes1": "",
```

```
    "transactionNotes2": "",
    "transactionNotes3": ""
  }
}
```

U. Update Account Data

```
{ path: "/Account/CreditCard", method: POST }
{
  "request": "updateAccount",
  "divisionId": "17",
  "paywayAccountToken": "10117024",
  "cardAccount":
  {
    "expirationDate": "07/25",
    "email": "UPDATED_dan.r.Coyote@gmail.com",
    "firstName": "UPDATED_Wiley",
    "lastName": "UPDATED_Coyote",
    "middleName": "UPDATED_Genius",
    "address": "UPDATED_12 Red Rock Terrace",
    "city": "UPDATED_Singing Desert",
    "state": "MA",
    "zip": "88888",
    "phone": "333-333-3333",
    "accountNotes1": "UPDATED_Here",
    "accountNotes2": "UPDATED_is my",
    "accountNotes3": "UPDATED_account"
  }
}
```


V. Update Transaction Notes

```
{ path: "/Payment/CreditCard", method: POST }
{
  "request": "updateTransactionNotes",

  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",
  "cardTransaction":
  {
    "sourceId": "20",
    "name": "<use transaction id of transaction>",
    "transactionNotes1": "Note 1 you want to insert"
    "transactionNotes2": "Note 2 you want to insert"
    "transactionNotes3": "Note 3 you want to insert"
  }
}
```

Reply

```
{
  "cardAccount": {
    "accountNotes1": "UPDATED_Here",
    "accountNotes2": "UPDATED_is my",
    "accountNotes3": "UPDATED_account",
    "accountNumber": "401288*****1881",
    "account_number_masked": "401288*****1881",
    "address": "UPDATED_12 Red Rock Terrace",
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "UPDATED_Singing Desert",
    "commercialCardType": 0,
    "divisionId": 1,
    "email": "UPDATED_dan.r.Coyote@gmail.com",
    "expirationDate": "0725",
    "firstFour": "4012",
    "firstName": "UPDATED_Wiley",
    "fsv": "",
    "inputMode": 1,
```

```
"lastFour": "1881",
"lastName": "UPDATED_Coyote",
"lastUsed": "1999-01-01 00:00:00-05",
"middleName": "UPDATED_Genius",
"onlinePaymentCryptogram": "",
"p2peInput": "",
"paywayToken": 10000130,
"phone": "3333333333",
"state": "MA",
"status": 2,
"zip": "88888"
},
"cardTransaction": {
  "addressVerificationResults": "N",
  "amount": 1000,
  "authorizationCode": "TAS642",
  "authorizedTime": "2021-02-25 00:00:00-05",
  "capturedTime": "2021-02-25 17:02:57-05",
  "cbMode": 4,
  "eciType": 1,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "name": "<use transaction id of transaction>",
  "pfpstatus": 3601,
  "pfpstatusString": "PFP Not Enabled",
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "00",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "Joe's Taxi 45th St",
  "referenceNumber": "",
  "resultCode": 0,
  "sessionToken_string": "0",
  "settledTime": "1999-01-01 00:00",
  "sourceId": 20,
  "status": 4,
```

```
"tax": 100,  
"testResultAVS": "",  
"testResultFSV": "",  
"transactionNotes1": "Updated Note1 ",  
"transactionNotes2": "Updated Note 2",  
"transactionNotes3": "Updated Note 3"  
},  
"paywayCode": "5000",  
"paywayMessage": ""  
}
```

W. Queue Authorize Hosted Request

path: /Payment/CreditCard method: POST

```
{  
  "cardTransaction": {  
    "amount": 1000,  
    "eciType": 1,  
    "sourceId": 20,  
    "name": "TEST-81-15",  
    "tax": 100,  
    "transactionNotes1": "",  
    "transactionNotes2": "",  
    "transactionNotes3": ""  
  },  
  "password": "1234567h",  
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",  
  "request": "queueAuthorize",  
  "userName": "villagePerson"  
}
```

queueAuthorize Reply

```
{  
  "paywayCode": "5000",  
  "paywayMessage": "",  
  "paywayRequestToken": "630CC360617453F46AE959BF815681AC",  
  "transactionName": ""  
}
```

}

X. Queue Sale Hosted Request

path: /Payment/CreditCard method: POST

```
{
  "cardTransaction": {
    "amount": 1000,
    "eciType": 1,
    "sourceId": 20,
    "name": "TEST-81-14",
    "tax": 100,
    "transactionNotes1": "",
    "transactionNotes2": "",
    "transactionNotes3": ""
  },
  "password": "1234567h",
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",
  "request": "queueSale",
  "userName": "villagePerson"
}
```

queueSale Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": "",
  "paywayRequestToken": "6D9DA56D2457C216FB38200DC3000738",
  "transactionName": ""
}
```

Y. Queue Trial Authorize

path: /Payment/CreditCard

method: POST

Request:{

"request": "queueTrialAuthorize",

"userName": "test",

"password": "1234567w",

"cardTransaction": {

"name": "TEST-107-15",

"sourceId": 11,

"transactionNotes1": "",

"transactionNotes2": "",

"transactionNotes3": ""

}

}

Reply:{

"paywayCode": "5000",

"paywayMessage": "",

"paywayRequestToken": "15F11209A8F57CF45D07FBBB2FC4DFBB",

"transactionName": ""

}

Z. Send A Queued Transaction

```
path: /Payment/CreditCard method: POST
{
  "accountInputMode": "primaryAccountNumber",
  "cardAccount": {
    "accountNotes1": "Here",
    "accountNotes2": "is my",
    "accountNotes3": "account",
    "accountNumber": "4012888888881881",
    "address": "12 Red Rock Terrace",
    "city": "Singing Desert",
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "07/19",
    "firstName": "Wiley",
    "fsv": "",
    "lastName": "Coyote",
    "middleName": "Genius",
    "phone": "222-222-2222",
    "state": "AZ",
    "zip": "01929"
  },
  "paywayRequestToken": "6D9DA56D2457C216FB38200DC3000738",
  "request": "sendQueuedTransaction"
}
```

```
sendQueuedTransaction Reply
{
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

}

AA. Retrieve Hosted Transaction Results

path: /Query/CreditCard method: POST

```
{
  "paywayRequestToken": "6D9DA56D2457C216FB38200DC3000738",
  "request": "hostedTransactionResults"
}
```

hostedTransactionResults Reply

```
{
  "paywayCode": "5000",
  "paywayMessage": "",

  "cardAccount": {
    "accountNotes1": "",
    "accountNotes2": "",
    "accountNotes3": "Info about account",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "0719",
    "firstFour": "4012",
    "firstName": "Wiley",
    "fsv": "",
    "divisionId": 17,
    "inputMode": 4,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": null,
    "middleName": "Genius",
  }
}
```

```
"p2peInput": "",
"paywayAccountToken": 10117024,
"phone": "2222222222",
"state": "AZ",
"status": 2,
"zip": "01929"
},
"cardTransaction": {
  "addressVerificationResults": "",
  "amount": 1000,
  "authorizationCode": "0987654321",
  "authorizedTime": "2020-04-30 00:00:00-04",
  "capturedTime": "2020-04-30 16:41:42-04",
  "cbMode": 2,
  "eciType": 1,
  "fraudSecurityResults": "",
  "fsvIndicator": "",
  "sourceId": 20,
  "name": "TEST-81-14",
  "pfpstatus": 0,
  "processorErrorMessage": "",
  "processorOrderId": "",
  "processorRecurringAdvice": "",
  "processorResponseDate": "",
  "processorResultCode": "",
  "processorSequenceNumber": 0,
  "processorSoftDescriptor": "",
  "referenceNumber": "123456",
  "resultCode": 5000,
  "sessionToken_string": "489",
  "settledTime": "1999-01-01 00:00",
  "status": 4,
  "tax": 100,
  "testResultAVS": "",
  "testResultFSV": "",
  "transactionNotes1": "",
  "transactionNotes2": "",
  "transactionNotes3": "",
```

“type”: <Varies depending on the queueing method>,

}

}

BB.Queue Add Account Hosted Request

path: /Account/CreditCard method: PUT

```
{  
  "divisionId": 17,  
  "password": "1234567h",  
  "paywaySessionToken": "85EAE7B823DFEA917E61610580794A97",  
  "request": "queueAddAccount",  
  "userName": "villagePerson"  
}
```

queueAddAccount Reply

```
{  
  "paywayCode": "5000",  
  "paywayMessage": "",  
  "paywayRequestToken": "34895B941F259821E328F74F19808865",  
  "transactionName": ""  
}
```

CC.Send Queued Add Account

```
path: /Account/CreditCard method: PUT
{
  "accountInputMode": "primaryAccountNumber",
  "cardAccount": {
    "accountNotes1": "Here",
    "accountNotes2": "is my",
    "accountNotes3": "account",
    "accountNumber": "4012888888881881",
    "address": "12 Red Rock Terrace",
    "city": "Singing Desert",
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "07/19",
    "firstName": "Wiley",
    "fsv": "",
    "divisionId": "17",
    "lastName": "Coyote",
    "middleName": "Genius",
    "phone": "222-222-2222",
    "state": "AZ",
    "zip": "01929"
  },
  "paywayRequestToken": "34895B941F259821E328F74F19808865",
  "request": "sendQueuedAddAccount"
}
```

```
sendQueuedAddAccount Reply
{
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

DD. Retrieve Hosted Add Account Results

```
path: /Query/CreditCard method: POST
{
  "paywayRequestToken": "34895B941F259821E328F74F19808865",
  "request": "hostedAddAccountResults"
}
```

```
hostedAddAccountResults Reply
{
  "paywayCode": "5000",
  "paywayMessage": ""

  "cardAccount": {
    "accountNotes1": "",
    "accountNotes2": "",
    "accountNotes3": "Info about account",
    "accountNumber": "401288*****1881",
    "address": "12 Red Rock Terrace",
    "addressList": null,
    "auLastUpdate": "1999-01-01 00:00:00-05",
    "auUpdateType": 0,
    "cardType": 1,
    "city": "Singing Desert",
    "commercialCardType": 0,
    "email": "dan.r.Coyote@gmail.com",
    "expirationDate": "0719",
    "firstFour": "4012",
    "firstName": "Wiley",
    "fsv": "",
    "divisionId": 17,
    "inputMode": 4,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": null,
    "middleName": "Genius",
    "p2peInput": "",
    "paywayAccountToken": 10117024,
    "phone": "2222222222",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  }
}
```

XV. Payway Test Response Results.

Payway provides common response results for the development and test/stage environments. Note that any account/transactions do not get transmitted to the credit network. The responses below are returned when specific accounts are used for processing.

WARNING: ONLY USE THE 'paywayCode' as the element to determine a transaction result.

Account Number	paywayCode	Result	status	Description
4212000098765437	5012	The transaction was declined.	8	Declined.
6011000992002133	5012	The transaction was declined.	8	Declined.
5499750000000007	5033	The transaction state was invalid.	7	Error/Rejected.
341111614241001	6000	Company Configuration Error.	7	Error/Rejected.
5432257101033879	6000	Other Error.-payway error 17. Payway does not have the mapping for this error.	7	Error/Rejected.
4723091948404774	5018	Payway cannot connect to the processor.	7	Error/Rejected.

4539406409079424	5004	Operation Prohibited- the service account does not have permission to process transactions.	7	Error/Rejected.
5334700378609508	5044	Invalid state or province code (CAN).	7	Error/Rejected.
4085025996546567	5045	Invalid zip or postal code(CAN).	7	Error/Rejected.
5490992902040454	5037	Invalid credit card expiration date.	7	Error/Rejected.
4716521635095344	5035	Invalid credit card account number.	7	Error/Rejected.
4024007101242989	5020	Duplicate transaction- the unique identifier was already submitted to payway for this source.	7	Error/Rejected.
4532755186829832	5049	Invalid amount- the amount cannot be \$0.00	7	Error/Rejected.
4916311551837608	5036	Invalid address.	7	Error/Rejected.
4916297024890714	5040	Invalid city.	7	Error/Rejected.
4351617035728045	5039	Invalid credit card method of payment-the merchant is not authorized to process this card type.	7	Error/Rejected.

4716930271374949	5032	Invalid transaction type- the customer's credit card issuer disallowed this type of transaction.	7	Error/Rejected.
4556445019882684	5009	The service account user has been set invalid.	7	Error/Rejected.
4532657650245018	5048	Invalid token-the token is invalid, closed or does not exist.	7	Error/Rejected.
4539824175193809	5038	The fraud security code is invalid- American Express 4 digits, all others 3 digits.	7	Error/Rejected.
4916174955547128	5003	Invalid password for the service account.	7	Error/Rejected.
4532456243248294	5051	Invalid sales tax.	7	Error/Rejected.
4556607866773403	5089	The reply from the credit card processor was invalid.	7	Error/Rejected