PaywayWS ACH Integration Guide

Date: September 30, 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.

Table of Contents

Revision History 5

Related Documentation 8

Introduction 8

A. Integration Process 9

1. Development (paywaywsdev.com) 9

2. Staging (paywaywstest.com) 9

3. Production (paywayws.com) 9

B. Server Locations 11

C. RESTful Resources 11

D. ACH Processing Differs from Credit Card Processing 12

E. Background vs. Interactive Requests 12

F. Payway Account Tokens 14

G. Background Requests 15

1. Get A Payway Session 15

2. Sale Request 16

3. Credit (Refund) 16

4. Void (Cancel) 17

5. Read Transaction Information 18

6. Read Account Information 18

7. Update Transaction Notes 19

H. Interactive Requests 20

1. Queue Transaction 21

2. Send Queued Transaction 22

3. Get Redirected Transaction Results 23

4. Queue Add Account Request 24

5. Send Queued Add Account Request 24

6. Get Redirected Add Account Results 25

I. Testing Your Integration 26

J. JSON Request Fields 26

1. Top Level 26

2. ACH Account 28

3. ACH Transaction 30

K. JSON Reply Fields 32

1. Top Level 32

2. ACHAccount 32

3. ACHTransaction 34

L. Payway Result Codes (paywayCode field) 36

M. JSON Examples 48

1. Logon Request 48

2. Background Requests 49

i. Add Account 49

ii. Sale with Payway Session and Primary Account Number 51

iii. Sale with One Time Session and Primary Account Number 54

iv. Sale with Token 57

v. Credit with Primary Account Number 60

vi. Credit with Token 63

vii. Void 66

viii. Update Transaction Notes NEW 68

ix. Queue Add Account Request 70

x. Send Queued Add Account Request 71

xi. Get Redirected Add Account Results 72

xii. Queue A Sale Transaction 73

xiii. Send Queued Transaction 74

xiv. Get Redirected Transaction Results 75

xv. Queue Add Account Request 77

xvi. Get Redirected Add Account Results 78

xvii. Retrieve Account Data with Token and Division Id 79

xviii. Retrieve Transaction Data (With Account Data) Using Name and Source Id 81

xix. Get Hosted Add Account Results 82

Revision History

Revision	Date	Initials	Comments
530	09/30/2020	DRN	Initial Draft
530	11/18/2020	DRF	Removed 'standardEntryClass', 'paymentType' and 'checkNumber' from ACH Account on page 23.
530	11/18/2020	DRF	Added standardEntryClass', 'paymentType' and 'checkNumber' to ACH Transaction on page 27.
530	11/18/2020	DRF	Removed 'standardEntryClass' and 'checkNumber' from the JSON example for 'AddAccount' request and reply on pages 42,43.
530	11/18/2020	DRF	Removed 'standardEntryClass' and 'checkNumber' from the JSON example for 'SendQueuedAddAccount' request on page 60.
530	11/18/2020	DRF	Removed 'standardEntryClass',paymentType' and 'checkNumber' from the JSON example for 'Payment/ACH' (bank Account) request and reply on pages 444,45,46.
530	11/18/2020	DRF	Added 'standardEntryClass','paymentType' and 'checkNumber' from the JSON example for 'Payment/ACH' (directDebitTransaction) request and reply on pages 44,45,46.
530	11/18/2020	DRF	Removed 'standardEntryClass',paymentType' and 'checkNumber' from the JSON example for

			‘Payment/ACH’ (bank Account) request and reply on pages 47,48,49.
530	11/18/2020	DRF	Added ‘standardEntryClass’,’paymentType’ and ‘checkNumber’ from the JSON example for ‘Payment/ACH’ (directDebitTransaction) request and reply on pages 47,48,49.
530	11/18/2020	DRF	Added ‘standardEntryClass’,’paymentType’ and ‘checkNumber’ from the JSON example for ‘Payment/ACH’ (directDebitTransaction) request and reply on pages 50,51.
530	11/18/2020	DRF	Removed ‘standardEntryClass’,paymentType’ and ‘checkNumber’ from the JSON example for ‘Payment/ACH’ (bank Account) request and reply on pages 50,51.
530	11/18/2020	DRF	Added ‘standardEntryClass’,’paymentType’ and ‘checkNumber’ from the JSON example for ‘Payment/ACH’ (directDebitTransaction) request and reply on pages 52,53,54.55,56,57,58
530	11/18/2020	DRF	Removed ‘standardEntryClass’,paymentType’ and ‘checkNumber’ from the JSON example for ‘Payment/ACH’ (bank Account) request and reply on pages 52,53,54,55,56,57,58.
530	11/18/2020	DRF	Removed ‘standardEntryClass’, and ‘checkNumber’ from ACH/Account on page 60.
530	11/18/2020	DRF	Removed ‘standardEntryClass’, and ‘checkNumber’ from ACH/Query on page 61.
530	11/18/2020	DRF	Added ‘standardEntryClass’,’paymentType’ and ‘checkNumber’ from the JSON example for ‘Payment/ACH’ (directDebitTransaction) on page 63.
530	11/18/2020	DRF	Removed ‘standardEntryClass’ and ‘checkNumber’ from the JSON example for ‘Payment/ACH’ (directDebitTransaction) on page 64.
530	11/18/2020	DRF	Removed ‘standardEntryClass’, and ‘checkNumber’ from ACH/Query (achaccount) on page 65,66.
530	11/18/2020	DRF	Added ‘standardEntryClass’,’paymentType’ and ‘checkNumber’ from the JSON example for ‘Query/ACH’ (achtransaction) request and reply on pages 65,66.

530	11/18/2020	DRF	Removed ‘standardEntryClass’, and ‘checkNumber’ from ACH/Query (achaccount) on page 68,69.
530	11/18/2020	DRF	Removed ‘standardEntryClass’, and ‘checkNumber’ from Payment/ACH (achaccount) on page 70,71.
530	11/18/2020	DRF	Added ‘standardEntryClass’,’paymentType’ and ‘checkNumber’ from the JSON example for ‘Payment/ACH’ (achtransaction) on page 70,71.
530	11/25/2020	DN	All references to terminalCIty and terminalState are removed.
531	2/1/2021	DRF	Corrected JSON reply message from ‘state’ to ‘status’ with the correct integer values on page 30.
533	3/11/2021	DRF	The ‘status’ table on page 30 was updated so that a status of 4 is Pending and status of 5 is Deposited.
533	3/11/2021	DRF	Updated and standardized convention for: sourceId is now sourceId for all references. divisionId is now divisionId for all references. companyId is now companyId for all references. **Note the old convention is backwards compatible EXCEPT new methods added after release 532**.
533	3/11/2021	DRF	Added support for updating transaction notes on page 17 and 63.
533	5/17/2021	DRF	Updated all API domains in Section ‘A.’

Related Documentation

- PaywayWS Credit Card 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.

This guide covers the integration of ACH (Automated Clearing House) payments which are direct debits and credits between your bank account and your customers' bank accounts. See PaywayWS Credit Card Integration Guide for integrating credit card and debit card payments.

A. 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.

1. Development (paywaywsdev.com)

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.

2. Staging (paywaywstest.com)

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.

3. Production (paywayws.com)

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.

B. 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

For example, to process a payment transaction on the development system the URL would be: https://paywaywsdev.com/PaywayWS/Payment/ACH.

C. 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/ACH	Create and manage transactions for ACH payments.
/Account/ACH	Create and manage ACH accounts including adding an account and acquiring the related token.
/Query/ACH	Retrieve detailed results for one or more ACH transactions or ACH accounts. Sensitive data is not returned or is returned masked.

D. ACH Processing Differs from Credit Card Processing

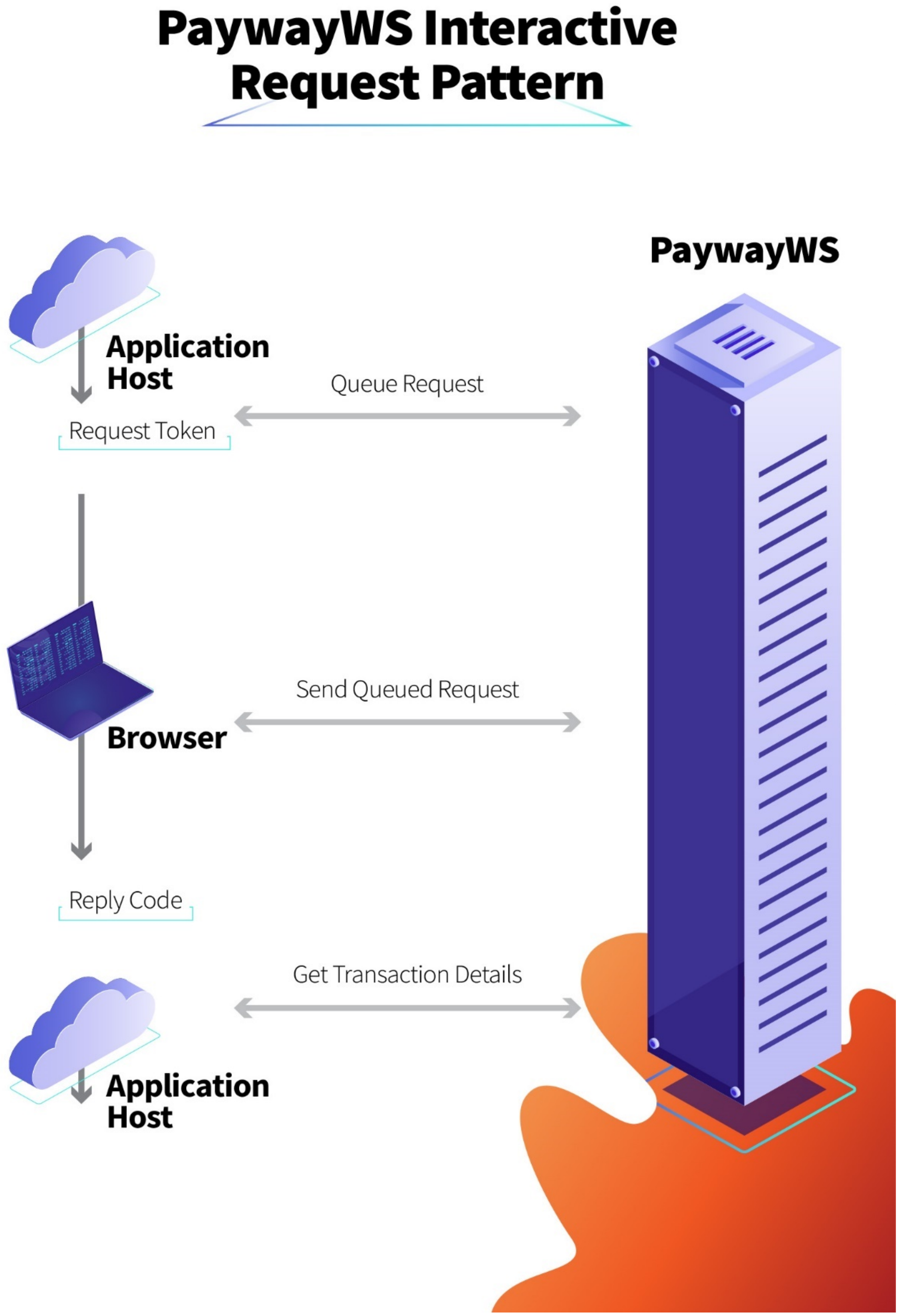
An ACH transaction differs from a credit card transaction in that the request is actually processed with the nightly batches, and there is no guarantee of funds being available. With a credit card authorization that is not the case; funds are guaranteed when the payment is authorized.   Think of an ACH request and electronically cashing a customer's check. However, because you avoid the credit card payment infrastructure, your costs for an ACH payment are a tiny fraction of what the same credit card transaction would be.

E. 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.



F. 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.

G. Background Requests

See the Examples section below for samples of all requests.

1. 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	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. Multiple active sessions for the same user are allowed. **DO NOT CALL THIS METHOD FROM A BROWSER**. This is to be called from a secure site connected to PaywayWS. User credentials for the development, testing, and live systems will be provided by Payway for integration and setup. The Payway_Admin web application is used by merchants to manage users and credentials.

2. Sale Request

JSON 'request' value	sale
DESCRIPTION	A sale request will cause funds to be deposited from the customer's bank account to the merchant's bank account.
URL	/Payment/ACH
METHOD	**Post**
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

3. Credit (Refund)

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

4. 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 customer's bank account or reverse a credit, refunding money to the merchant's bank account. Void will succeed on transactions that are captured, but not yet deposited. In general, voids must be done on the same day as the sale or credit. Otherwise a new request must be issued.
URL	/Payment/ACH
METHOD	**Post**
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

5. Read Transaction Information

JSON ‘request’ value	getTransaction
DESCRIPTION	Returns the transaction and payment account information for the specified transaction.
URL	/Query/ACH
METHOD	**Get**
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	Transactions are uniquely identified by a source ID and transaction name.

6. Read Account Information

JSON ‘request’ value	getAccount
DESCRIPTION	Used to find a particular payment account.
URL	/Query/ACH
METHOD	**Get**
URL PARAMS	None
CONSUMES	text/json

PRODUCES	text/json
NOTES	Payment accounts are uniquely identified by division ID and account token.

7. Update Transaction Notes

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

H. 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.

1. Queue Transaction

JSON ‘request’ value	queueSale or queuePreNote
DESCRIPTION	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. After queueSale the card page is loaded and sendQueuedTransaction is called via a java script.
URL	/Payment/ACH
METHOD	**Post**
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

2. Send Queued Transaction

JSON ‘request’ value	sendQueuedTransaction
DESCRIPTION	This request will send a previously queued transaction. Account information entered by the user is passed in this request.
URL	/Payment/ACH
METHOD	**Post**
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	The account data is sent from the browser via a java script when the user clicks on a particular button to make the payment. See the ACH Redirected Payment Form Integration Guide for more details.

3. Get Redirected 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/ACH
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.

4. 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/ACH
METHOD	**Post**
URL PARAMS	None
CONSUMES	text/json
PRODUCES	text/json
NOTES	

5. Send Queued Add Account Request

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

6. Get Redirected 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/ACH
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.

I. 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.

J. JSON Request Fields

1. Top Level

Name	Description
request	Request Name. This contains the request name, must be one of the following: “getPaywaySession” “addAccount” “sale” “credit” “void” “getTransaction” “getAccount” “queueSale” “sendQueuedTransaction” “redirectedTransactionResults” “queueAddAccount”

	“sendQueuedAddAccount” “hostedAddAccountResults”
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.
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.

2. ACH Account

Name	Description
firstName	First name from customer billing data. **Optional.**
middleName	Middle name. Usually blank. **Optional.**
lastName	Last name from customer billing data. **Required.**
address	Address from customer billing data, may include unit number, apartment, etc. **Optional.**
city	City from customer billing data. **Optional.**
state	State or province. **Optional.**
zip	Zip code (US) or province code. **Optional.**
phone	Customer phone number. **Optional.**
accountNumber	Customer account number. **Required.**
routingNumber	Routing number for the bank. **Required.**
bankName	Name of the bank.

	Optional.
AccountType	<blank> , 'checking' or 'savings'.
email	Email of the customer.
accountNotes1	Additional account information. **Optional.**
accountNotes2	Additional account information. **Optional.**
accountNotes3	Additional account information. **Optional.**

3. ACH Transaction

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>name</td><td>Unique name of the 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.</td></tr>
<tr><td>amount</td><td>Amount IN PENNIES. Ex $1.23 submitted as ‘123’
<b>Required.</b></td></tr>
<tr><td>tax</td><td>Tax amount IN PENNIES. Ex $1.23 submitted as ‘123’
<b>Tax cannot be more than the amount.</b>
<b>Optional. Can be ‘0’</b></td></tr>
<tr><td>standardEntryClass
<b>Required.</b></td><td><table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>PPD</td><td>Prearranged payments and deposits, most often on a recurring basis (pay-roll, bill payments)</td></tr>
<tr><td>TEL</td><td>Single debits authorized via telephone.</td></tr>
<tr><td>WEB</td><td>Single or recurring debits authorized via Internet.</td></tr>
<tr><td>CCD</td><td>Credits or debits where funds are distributed or consolidated between corporations.</td></tr>
</table></td></tr>
<tr><td>paymentType</td><td><b>Required for transactions with a SEC code of “TEL” or “WEB”.</b>
<b>Value: Space= Non-WEB Transaction</b>
<b>R= Recurring (Web only) S= Single (Web or TEL).</b></td></tr>
</table>

checkNumber	Customer check number. **Optional.**
testResult	**For Testing only. This value will be set as the payway return code.**
transactionNotes1	Additional 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. **Optional.**
transactionNotes2	Additional 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. **Optional.**
transactionNotes3	Additional 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. **Optional.**

K. JSON Reply Fields

1. Top Level

Name	Description
paywayCode	See the Payway Result Codes section below
paywayMessage	**Human readable message. NOT TO BE USED OR PROCESSED as a result indicator.**

2. ACHAccount

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.

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

lastFour	Last four digits of the customer bank account number.

3. ACHTransaction

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.

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>sourceId</td><td>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.</td></tr>
<tr><td>name</td><td>Unique name for the transaction.</td></tr>
<tr><td>resultCode</td><td>An integer value representing the result of the request.
See Section L for a complete listing of all codes.</td></tr>
<tr><td>status</td><td>
<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>
</table>
</td></tr>
<tr><td>authorizedTime</td><td>Time the transaction was authorized. Uninitialized if the transaction was not successful.</td></tr>
<tr><td>capturedTime</td><td>Time the transaction was captured. Uninitialized if the transaction was not successful.</td></tr>
<tr><td>settledTime</td><td>Time the transaction was posted for settlement. Not the same as the time the deposit is posted in the merchant’s account.</td></tr>
</table>

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.

L. 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	pay-wayCode	pay-wayMessage
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	**pay-wayCode**	**pay-wayMessage**
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.

Description	pay-wayCode	pay-wayMessage
PAYWAY_WS_ACCOUNT_CLOSED	5031	This account was marked closed by the account updater.
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.

Description	pay-wayCode	pay-wayMessage
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.
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.

Description	pay-wayCode	pay-wayMessage
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.
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.

Description	**pay-wayCode**	**pay-wayMessage**
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.
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.

Description	paywayCode	paywayMessage
PAYWAY_WS_INVALID_TRANSACTION_ID	5065	The transaction id specified is invalid.
PAYWAY_WS_INVALID_ONLINE_PAYMENT_CRYPTOGRAM	5066	The online payment cryptogram for 3D secure account input is invalid.
PAYWAY_WS_INVALID_REQUEST	5067	Invalid request type specified.
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.

Description	pay-wayCode	pay-wayMessage
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.
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.

Description	pay-wayCode	pay-wayMessage
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_INTERNAL_ERROR	6000	Internal Payway error. Contact Payway for assistance.

M. JSON Examples

1. Logon Request

```
path: /Session
method : POST
 Request : {
"request": "getPaywaySession",
“companyId": "0000000003",
   "password": "villagepwtest1!",
      "userName": "villagepwtest"
}

Reply:{
   "paywayCode": "5000",
   "paywayMessage": "",
   "paywaySessionToken": "7F9CD72CE4F4183795138EEAA0B152A8",
   "user": {
      "email": "mcollins@villagesoup.com",
      "firstName": "",
      "lastName": "",
      "name": "villagepwtest",
      "notes": "",
      "phone": "",
      "timeZone": "America/New_York"
   }
}
```

2. Background Requests

i. Add Account

```
path:       /Account/ACH
method:     PUT
Request:{
"request": "addAccount",
"paywaySessionToken":” 7F9CD72CE4F4183795138EEAA0B152A8",
“accountInputMode": "primaryAccountNumber",
    "bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "55878787",
    "accountType": "savings",
    "address": "12 Red Rock Terrace",
    "city": "Singing Desert",
    "email": "Richard@gmail.com",
    "firstName": "Richard",
    "lastName": "Lionheart",
    "middleName": "",
    "phone": "222-222-2222",
    "routingNumber": "909126789",
    "state": "AZ",
    "zip": "01929"
      },
    }

Reply:{
  "bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******8787",
    "accountType": "savings",
    "bankName": "",
    "city": "Singing Desert",
    "divisionId": 0,
    "email": "Richard@gmail.com",
    "firstName": "Richard",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "8787",
    "lastName": "Lionheart",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "",
    "paywayToken": 1200000207,
    "phone": "2222222222",
    "routingNumber": "909126789",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  },
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

ii. Sale with Payway Session and Primary Account Number

```
path:       /Payment/ACH
method:     POST
Request:{
"request": "sale",
"paywaySessionToken": "7F9CD72CE4F4183795138EEAA0B152A8",
"accountInputMode": "primaryAccountNumber",
"bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "99988878787",
    "accountType": "savings",
    "address": "12 Red Rock Terrace",
    "city": "Red Desert",
    "email": "Henry@gmail.com",
    "firstName": "Henry",
    "lastName": "VIII",
    "middleName": "",
   “phone": "111-222-2222",
    "routingNumber": "555126789",
    "state": "AZ"
    "zip": "01930"
  },
  "directDebitTransaction": {
    "amount": 1000,
    “checkNumber": "5432",
    "name": "TEST-ACH-026-00",
    "paymentType": "recurring",
     "standardEntryClass": "PPD"
    "sourceId": 11,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
}
```

```
Reply:{
  "bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******8787",
    "accountType": "unknown",
    "bankName": "",
    "city": "Red Desert",
    "divisionId": 7,
    "email": "Henry@gmail.com",
    "firstName": "Henry",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "8787",
    "lastName": "VIII",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "",
     "paywayToken": 1200000208,
    "phone": "1112222222",
    "routingNumber": "555126789",
    "state": "AZ",
    "status": 2,
    "zip": "01930"
  },
  "directDebitTransaction": {
    "amount": 1000,
    "authorizedTime": "2020-10-29 13:06:26-04",
    "capturedTime": "2020-10-29 13:06:26",
    "cbMode": 2,
     "checkNumber": "5432",
    "fsvIndicator": "",
    "id_deposit": 0,
    "name": "TEST-ACH-026-00",
    "paymentType": "recurring",
    "posting_date": "1999-01-01 00:00:00-05",
    "processorResultCode": "",
        "processorSoftDescriptor": "",
        "purpose": 0,
        "rejectReason": 0,
        "resultCode": 0,
        "sessionToken_string": "0",
        "settledTime": "1999-01-01 00:00:00-05",
        "sourceId": 11,
       _"standardEntryClass": "PPD"
        "standardEntryCode": 0,
        "status": 4,
        "tax": 100,
        "transactionNotes1": "Notes field 1",
        "transactionNotes2": "Order 123",
        "transactionNotes3": "Other note"
    },
    "paywayCode": "5000",
    "paywayMessage": ""
}
```

iii. Sale with One Time Session and Primary Account Number

```
path:       /Payment/ACH
method:     POST
Request:{
"request": "sale",
"userName": "villagepwtest",
"password": "villagepwtest1!",
 "accountInputMode": "primaryAccountNumber",
 "bankAccount": {
      "accountNotes1": "Notes about account",
      "accountNotes2": "More info",
      "accountNotes3": "",
      "accountNumber": "111111111",
      "accountType": "checking",
      "address": "12 Red Rock Terrace",
      "city": "Scrub Pine Desert",
      "email": "Wiley@gmail.com",
      "firstName": "Wiley",
      "lastName": "Coyote",
      "middleName": "Genius",
      "phone": "222-222-2222",
      "routingNumber": "123455555",
      "state": "AZ",
     "zip": "01932"
  },
  "directDebitTransaction": {
     "amount": 1000,
     "name": "TEST-ACH-026-01",
      "checkNumber": "5432",
     "paymentType": "single",
     "sourceId": 11,
      "standardEntryClass": "PPD",

     "tax": 100,
     "transactionNotes1": "Notes field 1",
     "transactionNotes2": "Order 123",
     "transactionNotes3": "Other note"
  },
}



Reply:{
  "bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******1111",
    "accountType": "checking",
    "bankName": "",
    "city": "Scrub Pine Desert",
    "divisionId": 7,
    "email": "Wiley@gmail.com",
    "firstName": "Wiley",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "1111",
    "lastName": "Coyote",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "Genius",
    "paywayToken": 1200000209,
     "phone": "2222222222",
    "routingNumber": "123455555",
    "state": "AZ",
    "status": 2,
    "zip": "01932"
  },
  "directDebitTransaction": {
    "amount": 1000,
    "authorizedTime": "2020-10-29 13:06:26-04",
    "capturedTime": "2020-10-29 13:06:27",
     "checkNumber": "5432",
    "cbMode": 2,
   "fsvIndicator": "",
    "id_deposit": 0,
    "name": "TEST-ACH-026-01",
    "paymentType": "single",
    "posting_date": "1999-01-01 00:00:00-05",
    "processorResultCode": "",
    "processorSoftDescriptor": "",
    "purpose": 0,
    "rejectReason": 0,
    "resultCode": 0,
    "sessionToken_string": "0",
    "settledTime": "1999-01-01 00:00:00-05",
    "sourceId": 11,
    "standardEntryClass": "PPD"
    "standardEntryCode": 0,
    "status": 4,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

iv. Sale with Token

```
path:       /Payment/ACH
method:     POST
Request:{
"request": "sale",
"paywayAccountToken": "1200000134",
"paywaySessionToken": "7F9CD72CE4F4183795138EEAA0B152A8",
 "accountInputMode": "paywayAccountToken",
 "directDebitTransaction": {
    "amount": 1000,
    “checkNumber”:”5432”,
    "name": "TEST-ACH-026-02",
    "paymentType": "single",
    "sourceId": 11,
    “standardEntryClass”:”PPD”,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
}

Reply:{
  "bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******1881",
    "accountType": "checking",
    "bankName": "Jackie Daytona Savings and Loan",
    "city": "Singing Desert",
    "divisionId": 7,
    "email": "dan.r.Coyote@gmail.com",
    "firstName": "Wiley",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "Genius",
    "paywayToken": 1200000134,
    "phone": "2222222222",
    "routingNumber": "123456789",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  },
  "directDebitTransaction": {
    "amount": 1000,
    "authorizedTime": "2020-10-29 13:06:27-04",
    "capturedTime": "2020-10-29 13:06:27",
    "cbMode": 2,
    “checkNumber”:”5432”;
    "fsvIndicator": "",
    "id_deposit": 0,
    "name": "TEST-ACH-026-02",
    "paymentType": "unknown",
    "posting_date": "1999-01-01 00:00:00-05",
    "processorResultCode": "",
    "processorSoftDescriptor": "",
    "purpose": 0,
    "rejectReason": 0,
    "resultCode": 0,
    "sessionToken_string": "0",
    "settledTime": "1999-01-01 00:00:00-05",
    "sourceId": 11,
    “standardEntryClass”:”PPD”,
    "standardEntryCode": 0,
    "status": 4,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

v. Credit with Primary Account Number

```
path:       /Payment/ACH
method:     POST
Request:{
"request": "credit",
"paywaySessionToken": "7F9CD72CE4F4183795138EEAA0B152A8",
"accountInputMode": "primaryAccountNumber",
 “bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "55487263",
    "accountType": "checking",
    "address": "12 Red Rock Terrace",
    "bankName": "Jean Louis' Maison D'Or",
    "city": "Road Runner Desert",
    "email": "Hamlet@gmail.com",
    "firstName": "Hamlet",
    "lastName": "Denmark",
    "middleName": "",
    "phone": "777-222-2222",
    "routingNumber": "444488857",
    "state": "AZ",
    "zip": "01933"
  },
  "directDebitTransaction": {
    "amount": 1000,
   “checkNumber”:”5432”,
   “standardEntry Class”:”PPD”,
    "name": "TEST-ACH-026-03",
    "paymentType": "recurring",
    "sourceId": 11,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
}

Reply:{
  "bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******7263",
    "accountType": "checking",
    "bankName": "Jean Louis' Maison D'Or",
    "city": "Road Runner Desert",
    "divisionId": 7,
    "email": "Hamlet@gmail.com",
    "firstName": "Hamlet",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "7263",
    "lastName": "Denmark",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "",
    "paywayToken": 1200000210,
    "phone": "7772222222",
    "routingNumber": "444488857",
    "state": "AZ",
    "status": 2,
    "zip": "01933"
  },
  "directDebitTransaction": {
    "amount": 1000,
    "authorizedTime": "2020-10-29 13:06:27",
    "capturedTime": "2020-10-29 13:06:27",
    "cbMode": 2,
    “checkNumber”:”5432”,
    "fsvIndicator": "",
    "id_deposit": 0,
    "name": "TEST-ACH-026-03",
    "paymentType": "recurring",
    "posting_date": "1999-01-01 00:00:00-05",
    "processorResultCode": "",
    "processorSoftDescriptor": "",
    "purpose": 0,
    "rejectReason": 0,
    "resultCode": 0,
    "sessionToken_string": "0",
    "settledTime": "1999-01-01 00:00:00-05",
    "sourceId": 11,
    “standardEntryClass”:”PPD”,
    "standardEntryCode": 0,
    "status": 4,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

vi. Credit with Token

```
path:       /Payment/ACH
method:     POST
Request:{
"request": "credit",
"paywayAccountToken": "1200000134",
"paywaySessionToken": "7F9CD72CE4F4183795138EEAA0B152A8",
"accountInputMode": "paywayAccountToken",
"directDebitTransaction": {
    "amount": 1000,
    “checkNumber”:”5432”,
    "name": "TEST-ACH-026-04",
    "paymentType": "recurring",
    "sourceId": 11,
    “standardEntryClass”:”PPD”,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
}

Reply:{
  "bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******1881",
    "accountType": "checking",
    "bankName": "Jackie Daytona Savings and Loan",
    "city": "Singing Desert",
    "divisionId": 7,
    "email": "dan.r.Coyote@gmail.com",
    "firstName": "Wiley",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "Genius",
    "paywayToken": 1200000134,
    "phone": "2222222222",
    "routingNumber": "123456789",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  },
  "directDebitTransaction": {
    "amount": 1000,
    "authorizedTime": "2020-10-29 13:06:27",
    "capturedTime": "2020-10-29 13:06:27",
    "cbMode": 2,
    “checkNumber”:”5432”,
    "fsvIndicator": "",
    "id_deposit": 0,
    "name": "TEST-ACH-026-04",
    "paymentType": "recurring",
    "posting_date": "1999-01-01 00:00:00-05",
    "processorResultCode": "",
    "processorSoftDescriptor": "",
    "purpose": 0,
    "rejectReason": 0,
    "resultCode": 0,
    "sessionToken_string": "0",
    "settledTime": "1999-01-01 00:00:00-05",
    "sourceId": 11,
    “standardEntryClass”:”PPD”,
    "standardEntryCode": 0,
    "status": 4,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
  "paywayCode": "5000",
   "paywayMessage": ""
}
```

vii. Void

```
path:       /Payment/ACH
method:     POST
Request:{
 "request": "void",
 "paywaySessionToken":
"7F9CD72CE4F4183795138EEAA0B152A8",
 "directDebitTransaction": {
     "name": "TEST-ACH-026-04",
     "sourceId": 11
   },
}

Reply:{
   "bankAccount": {
     "accountNotes1": "Notes about account",
     "accountNotes2": "More info",
     "accountNotes3": "",
     "accountNumber": "******1881",
     "accountType": "checking",
     "bankName": "Jackie Daytona Savings and Loan",
     "city": "Singing Desert",
     "divisionId": 7,
     "email": "dan.r.Coyote@gmail.com",
     "firstName": "Wiley",
     "inputMode": 1,
     "key_alias": 60,
     "lastFour": "1881",
     "lastName": "Coyote",
     "lastUsed": "1999-01-01 00:00",
     "middleName": "Genius",
     "paywayToken": 1200000134,
     "phone": "2222222222",
     "routingNumber": "123456789",
     "state": "AZ",
     "status": 2,
     "zip": "01929"
```

```
  },
  "directDebitTransaction": {
    "amount": 1000,
    "authorizedTime": "2020-10-29 13:06:27-04",
    "capturedTime": "2020-10-29 13:06:27-04",
    "cbMode": 2,
    "checkNumber": "",
    "fsvIndicator": "",
    "id_deposit": 0,
    "name": "TEST-ACH-026-04",
    "paymentType": "recurring",
    "posting_date": "1999-01-01 00:00:00-05",
    "processorResultCode": "",
    "processorSoftDescriptor": "",
    "purpose": 0,
    "rejectReason": 0,
    "resultCode": 0,
    "sessionToken_string": "0",
    "settledTime": "1999-01-01 00:00:00-05",
    "sourceId": 11,
    "standardEntryClass": "PPD",
    "standardEntryCode": 0,
    "status": 6,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

viii. Update Transaction Notes NEW

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

      "directDebitTransaction" :
      {
                  "sourceId" : "3",
                  "name"     : "20210225_01_ACH_SaleWithToken",

                  "transactionNotes1" : "Updated 1",
                  "transactionNotes2" : "Updated 2",
                  "transactionNotes3" : "Updated 3"
       }
}
```

Reply

```
{
   "bankAccount": {
      "accountNotes1": "",
      "accountNotes2": "",
      "accountNotes3": "",
      "accountNumber": "******7743",
      "accountType": "checking",
      "address": "123",
      "bankName": "",
      "city": "Anytown",
      "divisionId": 1,
      "email": "me@me.com",
      "firstName": "Wilma",
      "inputMode": 1,
      "key_alias": 60,
      "lastFour": "7743",
      "lastName": "Flintstone",
      "lastUsed": "1999-01-01 00:00",
      "middleName": "",
      "paywayToken": 1200000024,
        "phone": "1234567890",
        "routingNumber": "111111118",
        "state": "FL",
        "status": 2,
        "zip": "01234"
    },
    "directDebitTransaction": {
        "amount": 1000,
        "authorizedTime": "2021-03-02 11:36:39-05",
        "capturedTime": "2021-03-02 11:36:39-05",
        "cbMode": 4,
        "checkNumber": "5432",
        "fsvIndicator": "",
        "id_deposit": 0,
        "name": "20210225_01_ACH_SaleWithToken",
        "paymentType": "single",
        "posting_date": "1999-01-01 00:00:00-05",
        "processorResultCode": "",
        "processorSoftDescriptor": "",
        "purpose": 0,
        "rejectReason": 0,
        "resultCode": 0,
        "sessionToken_string": "0",
        "settledTime": "1999-01-01 00:00:00-05",
        "sourceId": 3,
        "standardEntryClass": "PPD",
        "status": 4,
        "tax": 100,
        "terminalCity": "",
        "terminalState": "",
        "transactionNotes1": "Updated 1",
        "transactionNotes2": "Updated 2",
        "transactionNotes3": "Updated 3"
    },
    "paywayCode": "5000",
    "paywayMessage": ""
}
```

ix. Queue Add Account Request

```
path:       /Account/ACH
method:     PUT
Request:{
"request": "queueAddAccount",
"userName": "villagepwtest",
"password": "villagepwtest1!",
"companyId": "3",
"divisionId": 7,
}

Reply:{
  "paywayCode": "5000",
  "paywayMessage": "",
  "paywayRequestToken":
"13F5E41CAA621DCC047A747CC6D18D39",
  "transactionName": "0000000000"
}
```

x. Send Queued Add Account Request

```
path:        /Account/ACH
method:    PUT
Request:{
"request": "sendQueuedAddAccount",
"paywayRequestToken":
"13F5E41CAA621DCC047A747CC6D18D39",
"accountInputMode": "primaryAccountNumber",
  "bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "5542965",
    "accountType": "checking",
    "address": "12 No Rain Terrace",
    "city": "Singing Desert",
    "email": "lois@gmail.com",
    "firstName": "Lois",
    "lastName": "Lane",
    "middleName": "",
    "phone": "546-654-3278",
    "routingNumber": "765488857",
    "state": "AZ",
    "zip": "01929"
  },
}

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

xi. Get Redirected Add Account Results

path: /Query/ACH
method: POST

```
Request:{
"request": "redirectedAddAccountResults",
"paywayRequestToken":
"2FDBFA54D4855AC62D92E4A928E3C6A9",
}

Reply:{
  "achaccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******8999",
    "accountType": "checking",
    "bankName": "",
    "city": "Singing Desert",
    "divisionId": 7,
    "email": "dave@gmail.com",
    "firstName": "Dave",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "8999",
    "lastName": "MacBeth",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "",
    "paywayToken": 10000227,
    "phone": "8882222222",
    "routingNumber": "023458712",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  },
  "achtransaction": null,
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

xii. Queue A Sale Transaction

path: /Payment/ACH

method: POST

```
Request:{
"request": "queueSale",
“userName": "villagepwtest",
 "password": "villagepwtest1!",
 "companyId": "3",
 "directDebitTransaction": {
      "amount": 1000,
      "name": "TEST-ACH-026-05",
      "checkNumber": "5432",
      "paymentType": "recurring",
      "sourceId": 11,
     "standardEntryClass": "PPD",
      "tax": 100,
      "transactionNotes1": "Queued",
      "transactionNotes2": "This one",
      "transactionNotes3": ""
   },
}

Reply:{
   "paywayCode": "5000",
   "paywayMessage": "",
   "paywayRequestToken":
"888F715B12FE8542D76C26009CE80B41",
   "transactionName": ""
}
```

xiii. Send Queued Transaction

```
path:       /Payment/ACH
method:   POST
Request:{
"request": "sendQueuedTransaction",
 “paywayRequestToken":
"888F715B12FE8542D76C26009CE80B41",
"accountInputMode": "primaryAccountNumber",
"bankAccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "55487263",
    "accountType": "checking",
    "address": "12 Broken Rock Terrace",
    "city": "Singing Desert",
    "email": "propsero@gmail.com",
    "firstName": "Prospero",
    "lastName": "Koi",
    "middleName": "Genius",
    "phone": "222-786-2222",
    "routingNumber": "890488857",
    "state": "AZ",
    "zip": "01942"
  },
}

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

xiv. Get Redirected Transaction Results

```
path:       /Query/ACH
method:     POST
Request:{
"request": "redirectedTransactionResults",
"paywayRequestToken":
"6AB30FA1C25D3CC3E331BA47EA674190",
}
Reply:{
  "achaccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******7263",
    "accountType": "checking",
    "bankName": "",
    "city": "Singing Desert",
    "divisionId": 7,
    "email": "propsero@gmail.com",
    "firstName": "Prospero",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "7263",
    "lastName": "Koi",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "Genius",
    "paywayToken": 1200000220,
    "phone": "2227862222",
    "routingNumber": "890488857",
    "state": "AZ",
    "status": 2,
    "zip": "01942"
  },
  "achtransaction": {
    "amount": 1000,
    "authorizedTime": "2020-10-29 13:06:28-04",
    "capturedTime": "2020-10-29 13:06:28-04",
    "cbMode": 2,
    "checkNumber": "",
    "fsvIndicator": "",
    "id_deposit": 0,
    "name": "TEST-ACH-026-05",
    "paymentType": "recurring",
    "posting_date": "1999-01-01 00:00:00-05",
    "processorResultCode": "",
    "processorSoftDescriptor": "",
    "purpose": 0,
    "rejectReason": 0,
    "resultCode": 5000,
    "sessionToken_string": "0",
    "settledTime": "1999-01-01 00:00:00-05",
    "sourceId": 11,
    "standardEntryClass": "PPD",
    "standardEntryCode": 0,
    "status": 4,
    "tax": 100,
    "transactionNotes1": "Queued",
    "transactionNotes2": "This one",
    "transactionNotes3": ""
  },
  "paywayCode": "5000",
  "paywayMessage": ""
}
}
```

xv. Queue Add Account Request

```
path:       /Account/ACH
method:     PUT
Request:{
"request": "queueAddAccount",
"userName": "villagepwtest",
"password": "villagepwtest1!",
 "companyId": "3",
"divisionId": 7,
}

Reply:{
   "paywayCode": "5000",
   "paywayMessage": "",
   "paywayRequestToken":
"13F5E41CAA621DCC047A747CC6D18D39",
   "transactionName": "0000000000"
}
```

xvi. Get Redirected Add Account Results

path: /Query/ACH

method: POST

```
Request:{
"request": "redirectedAddAccountResults",
"paywayRequestToken":
"C61AD8BB743D9E3BF9EF4FED0B10E448",
}

Reply:{
  "achaccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******2965",
    "accountType": "checking",
    "bankName": "",
    "city": "Singing Desert",
    "divisionId": 7,
    "email": "lois@gmail.com",
    "firstName": "Lois",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "2965",
    "lastName": "Lane",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "",
    "paywayToken": 10000228,
    "phone": "5466543278",
    "routingNumber": "765488857",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  },
  "achtransaction": null,
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

xvii. Retrieve Account Data with Token and Division Id

```
path:      /Query/ACH
method:    POST
Request:{
"request": "getAccount",
"divisionId": "7",
“paywayAccountToken": "1200000134",
“paywaySessionToken": "7F9CD72CE4F4183795138EEAA0B152A8",
}

Reply:{
  "achaccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******1881",
    "accountType": "checking",
    "bankName": "Jackie Daytona Savings and Loan",
    "city": "Singing Desert",
    "divisionId": 7,
    "email": "dan.r.Coyote@gmail.com",
    "firstName": "Wiley",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "Genius",
    "paywayToken": 1200000134,
    "phone": "2222222222",
    "routingNumber": "123456789",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
},
"achtransaction": null,
"paywayCode": "5000",
"paywayMessage": ""
```

xviii. Retrieve Transaction Data (With Account Data) Using Name and Source Id

```
path:       /Query/ACH
method:     POST
Request:{
"request": "getTransaction",
"paywaySessionToken": "7F9CD72CE4F4183795138EEAA0B152A8",
"sourceId": "11",
"transactionName": "TEST-ACH-026-04"
}

Reply:{
  "achaccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******1881",
    "accountType": "checking",
    "bankName": "Jackie Daytona Savings and Loan",
    "city": "Singing Desert",
    "divisionId": 7,
    "email": "dan.r.Coyote@gmail.com",
    "firstName": "Wiley",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "1881",
    "lastName": "Coyote",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "Genius",
    "paywayToken": 1200000134,
    "phone": "2222222222",
    "routingNumber": "123456789",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  },
  "achtransaction": {
    "amount": 1000,
    "authorizedTime": "2020-10-29 13:06:27-04",
    "capturedTime": "2020-10-29 13:06:27-04",
    "cbMode": 2,
    “checkNumber": "5432",
    "fsvIndicator": "",
    "id_deposit": 0,
    "name": "TEST-ACH-026-04",
    "paymentType": "recurring",
    "posting_date": "1999-01-01 00:00:00-05",
    "processorResultCode": "",
    "processorSoftDescriptor": "",
    "purpose": 0,
    "rejectReason": 0,
    "resultCode": 5000,
    "sessionToken_string": "0",
    "settledTime": "1999-01-01 00:00:00-05",
    "sourceId": 11,
    "standardEntryClass": "PPD",
    "standardEntryCode": 0,
    "status": 6,
    "tax": 100,
    "transactionNotes1": "Notes field 1",
    "transactionNotes2": "Order 123",
    "transactionNotes3": "Other note"
  },
  "paywayCode": "5000",
  "paywayMessage": ""
}
```

xix. Get Hosted Add Account Results

```
path:       /Query/ACH
method:     POST
Request:{
  "paywayRequestToken":
"208651653FD0EBA11B663016C3D276E1",
  "request": "redirectedAddAccountResults"
}

Reply:{
  "achaccount": {
    "accountNotes1": "Notes about account",
    "accountNotes2": "More info",
    "accountNotes3": "",
    "accountNumber": "******8999",
    "accountType": "checking",
    "address": "12 Big Rock Terrace",
    "bankName": "",
    "city": "Singing Desert",
    "divisionId": 7,
    "email": "dave@gmail.com",
    "firstName": "Dave",
    "inputMode": 1,
    "key_alias": 60,
    "lastFour": "8999",
    "lastName": "MacBeth",
    "lastUsed": "1999-01-01 00:00",
    "middleName": "",
    "paywayToken": 10000271,
    "phone": "8882222222",
    "routingNumber": "023458712",
    "state": "AZ",
    "status": 2,
    "zip": "01929"
  },
  "achtransaction": null,
  "paywayCode": "5000",
  "paywayMessage": ""
}
```