PaySAFE

API Documentation

API Version: 1.5.2
August, 2018

Service Summary

PaySAFE is an online closing table that brings together any buyer and seller looking for a safe, secure way to complete high value transactions. Either party can draft a purchase agreement with unique terms for each transaction.

Buyers then transfer the full purchase amount into PaySAFE's neutral, third party escrow account - locking in the sale and verifying funds up front. Sellers are then left to complete the contract terms as agreed upon by both parties and request payment. Buyers then agree to a release of funds and PaySAFE transfers the entire agreed upon amount - minus PaySAFE's fee - into the seller's specified bank account.

A complete record of communication between both parties that occurs within the PaySAFE platform. This record is also kept on file and is available 24/7 to either party.

There is one notable difference between how PaySAFE works and how a transactions close with a traditional payment processor. Typically with the latter, the transaction settlement would be completed in a few minutes. Our service on the other hand is more akin to an online closing table. It is targeted for transaction of high dollar values where the buyer is unwilling to blindly send the seller money without some type of risk mitigation/protection. Our service is most valuable to customers at the dollar amount where traditional payment systems stop offering protection. Along with auditing/documenting the complete transaction, PaySAFE holds the funds in escrow until both parties agree to a release. As such, transactions can span days. Our workflow is as follows:

1. Transaction is created. (Typically by the seller. In the case of API integration, whoever listed the item would be the creator and have rights to edit the terms of the contract.)
2. Creator invites the other party (Invitee is typically the buyer.)
3. The invitee joins the transaction by following a link within an email sent to them. They have the choice to accept/reject the terms of the contract at this point. Note that is possible to start the transaction in a buyer/seller pre-approval state shortening the workflow.

4. Once both parties have agreed to the terms, the buyer is asked by PaySAFE to fund the transaction.
5. Once PaySAFE has verified the funds to be legitimate (and complete Know Your Customer verification for both the buyer and seller), the transaction is set to “active” and the seller is instructed to complete the terms of the transaction (normally ship the goods to the buyer).
6. Once the buyer receives the goods (or generally speaking the "terms of release" outlined in the contract are met), PaySAFE releases the funds held in escrow to the seller and the transaction is moved to the closed status.

Currently, using the API to integrate with us shortens the workflow to where the creator moves directly into a transaction that is waiting for him to invite the other party once s/he has verified that the contract terms are appropriate. It is possible for the integrator to accept the responsibility of gaining approvals from the buyer and seller and starting the workflow at the point where the buyer must fund the transaction.

As with the traditional payment gateway providers that notify the integrator of the status of the payment, the PaySAFE API can be used in a similar fashion to notify the integrator of transaction status changes. As mentioned above, however, transactions initiated with PaySAFE go through a workflow with many possible status changes from start and close. A complete summary of all possible status’ are included at the end of this document.

*__Important Note__: PaySAFE currently restricts transactions in the following states: California, Texas, Washington, Nevada, Arizona, Idaho, New Mexico, Oregon, Montana, Utah, or Vermont. Please note that this list can and will be modified as conditions change. Accruit/PaySAFE will notify partners as changes occur.

Attempting to create a user in one of these states, or attempting to create a transaction with a buyer/seller/creator in any of these states will return a message that the service is not available in that state (see specific API JSON scripting examples).

Security / Authentication

In our production environment, PaySAFE only listens to web traffic on the default HTTPS port 443. Any non-secure requests are simply ignored. When an integrator is making RESTful calls to the API, Basic HTTP Authentication is used on each call.

To implement Basic Http Authentication, one simply adds an “Authorization” header to the HTTPS request. The value of the authorization header is a Base64 encoded string containing the username and password separated by a colon.

Example:

```
Authorization: Basic VXN1ciBOYW1101Bhc3N3b3Jk
```

In the example above the Base64 encoding translates to "*User Name:Password"*.

Integrators should use their partner code as the username, and their API key as the password.

Both of these will be provided by your PaySAFE contact.

PaySAFE maintains separate API keys for the test and production environments for each of its partners. Make sure to use the correct credentials when contacting the respective servers.

API Endpoints

PaySAFE HTTPS endpoints through which integrators can create users, create and cancel transactions, and other useful actions.

```
https://api.paysafeescrow.com/api/v1/user
https://api.paysafeescrow.com/api/v1/transaction
https://api.paysafeescrow.com/api/v1/transaction/123/cancel
```

Note that these are the production endpoints. Developers can access test versions of these endpoints by replacing the hostname to ***APItest.paysafeescrow.com***.

Make sure to set both the **Accept** and **Content-Type** http headers to ***application/json.*** Most REST client libraries will do this for you automatically.

POST /api/v1/user

A POST to the user endpoint to create a new user. This method will create a user and return an identifier which can be used later when creating transactions.

In order to create transactions, you need at least one PaySAFE user to act as the transaction owner. Furthermore, if you would like the transaction to start in any status past "Estimate", then you will need to have a user Id for both the buyer and seller

Note that if a user with the same email address already exists, a duplicate is not created. Instead, the identifier of the existing user is returned. Furthermore, the user details on record are maintained and the user demographics provided in the request are ignored.

When POSTing to this endpoint, the body of the HTTP request should contain a json object with the following properties:

Property	Description	Max Length
AccountName	Business name or combination of first and last name	100

FirstName	user's first name	100
LastName	user's last name	100
Phone	user's phone number	20
Email	user's email address and future PaySAFE login	253
TrustLevel	0: No validation of contact information 1: email verified 2: includes phone verification 3. includes address/name verified Note that each partner has an assigned max trust level that can be used. By default, only 0 and 1 are allowed. (optional – default = 0)	N/A
TrackingCode	This can be used to segment users by the integrator. For example, an integrator may assign a tracking code per sales rep. **(optional)**	45
BirthDate	User's date of birth	N/A
Address	user's address	150
City	user's city	50
State	user's state / province	50
PostalCode	The transaction owner's billing zip/postal code	15
Country	2-letter country code. Examples: US-USA CA-Canada AU- Australia.	2

Possible HTTP Responses to /api/v1/user:

HTTP Status Code	Description
OK (200)	The expected result of all api calls. This indicates that everything worked as expected. The BODY of the http response in this case simply contains the new user's Identifier.

Bad Request (400)	This response is used anytime there is validation errors in the request. The response body will include a json object identifying the field names that caused the validation errors and a message indicating what is wrong.
	Example response:
	`[`
	`{ "field":"request.AccountName",`
	`"message":"The AccountName field is required."`
	`},`
	`{ "field":"request.State",`
	`"message":"The State field is required."`
	`},`

	{"State":[
	"The State must be 2 characters long."
	]
	}
	]
	Field validation will respond if a field submitted is longer than the field length in the API. For instance, if an attempt is made to create a user with an account name of longer than 100 characters, the following response will be sent:
	{
	"AccountName": [
	"The field AccountName must be a string or array type with a maximum length of '100'."
	]
	}
	Additionally, if a call attempts to create a user in a state where PaySAFE has restricted activity (see note in Service Overview), the response will be:
	"Service is not available in <State>" (where <State> is the state in which the created user resides.)

Server Error (500)	This response indicates a situation where the PaySafe API cannot recover from an error

POST /api/v1/transaction

A POST to the *transaction* endpoint creates a new transaction. This method will return an identifier to this newly created transaction. Emails will be sent to the buyer and seller informing them of their participation in the transaction and providing a direct link to it. If the user(s) have also been created via the user API call, then they will first be asked to select a password.

Property	Description	Max Length
Type	Possible Values: **[Goods | Auto | Livestock]** For any non-general "Goods" types, make sure that the TemplateData property is populated with the correct template data. More details follow.	N/A
Status	Possible Values: **[Estimate | PendingApproval | EstimateApproved]** Estimates are only visible to the user assigned as the creator. Only the creator needs to be identified, but the invitee can be optionally set.	N/A

	PendingApproval indicates that the creator is happy with the contract and is awaiting the other party to agree. Both the buyer and seller are required to be provided. EstimateApproved indicates that the creator has proxied both buyer and seller approval to the transaction details.	
Name	The name of the transaction. This name is how the transaction will be displayed when viewing a list or summary of transactions.	128
PromoCode	This is an **optional** promotional code for special pricing.	16
TrackingCode	This is an **optional** tracking code intended to allow for transaction segmentation (if desired by the integrator). Some partners use this to track conversion rates.	45
Description	The basic terms of the transaction. Typically describes the good or service being traded.	2048
TemplateData	For any transactions created with a type other than “Goods”, this property should hold a json string containing the template’s properties and values. Details of these templates are included below. **(Optional)**	2048
ReleaseTerms	Often buyers and sellers have agreed to specific terms of release. If available, use this field for that purpose. An example would be, “Upon receipt of shipping notification”, etc. **(Optional)**	1024
ClassifiedAdUrl	The fully qualified URL that would direct a user to see the original listing. If provided, this link becomes part of the transaction details. **(Optional)**	256
CreatorID	The PaySAFE User ID for the individual who owns the transaction.	N/A
BuyerID	The buyer’s PaySAFE User ID. **(Optional)**	N/A
SellerID	The seller’s PaySAFE User ID. **(Optional)**	N/A
BuyerInvitationEmail	If no PaySAFE user is identified as the buyer, it is possible to pass only their email. Not that at	253

	least the creator needs to be an existing PaySAFE user identified by their UserID. This property is only available if creating transactions in the “Estimate” status. **(Optional)**	
SellerInvitationEmail	If no PaySAFE user is identified as the seller, it is possible to pass only their email. Not that at least the creator needs to be an existing PaySAFE user identified by their UserID. This property is only available if creating transactions in the “Estimate” status. **(Optional)**	253
SalePrice	SalesPrice=ContractValue. Typically the sale price of the goods being exchanged. This is the amount that will be required to be deposited into escrow.	N/A
ExternalID	If the transaction being created is linked to an item internal to the integrator’s (partner) system, it is possible to link the item to the transaction being created through this field. This property can be used to identify the goods/service being traded on PaySAFE. It can be any value up to 64 bytes long. **(Optional)**	64
CallbackUrl	If provided, this fully qualified URL will be hit with an HTTP POST on all transaction events that do not violate PaySAFE privacy policies. See section below for details. **(Optional)**	512
List of Payment Items (see object definition of Payment Item below)	A list of all payments for this transaction	N/A
Origin	String containing origin of call. For any API calls, the value is “api”.	45
InspectionHours	Hours to hold the payment schedule within PaymentRequested status (50). Once the inspection hours have expired, the transaction’s payment schedule will be approved. Upon expiration, the transaction status is advanced to PaymentApproved status (70).	N/A
AutoRequestPayment	Authorized the transaction to move status automatically from Active (40) to PaymentRequested (50) upon reaching Active status.	N/A

PartnerCode	Unique identifier assigned by PaySAFE to track activity and reporting.	45
AutoApprovePayment	Authorized transaction to move status automatically from PaymentRequested (50) to PaymentApproved (70) upon reaching PaymentRequested status. (You cannot have both InspectionHours and AutoApprovePayment selected).	N/A

Template Data: As mentioned above, PaySAFE does support other more specific transaction types and additional metadata can be included if desired by selecting the appropriate transaction Type and populate the TemplateData field with the particulars for the chosen template. . Currently 2 other types of transactions are supported. **Livestock** and **Auto**. If these type of transactions are targeted, the following properties should be included as a json string and placed in the *TemplateData* property.

Livestock

Parameter	Value	Description
Breed	string	This field is used to identify the type of animal being sold. Examples would be: Angus, Devon, Gloucester, etc. **(Optional)**
HeadCount	string	Typical a number, but any value can be used to describe the quantity of animals being traded. **(Optional)**

Auto

Parameter	Value	Description
Make	string	Make of the vehicle being sold. **(Optional)**
Model	string	Vehicle model. **(Optional)**
VIN	string	Unique serial number of the vehicle. **(Optional)**
Odometer	string	Miles/kilometers or hours of operations. **(Optional)**
Year	string	Year the vehicle was manufactured.**(Optional)**

Payment Item

Parameter	Value	Description
Label	string	Payment Label. **(Optional)**
Member	string	“Creator”, “Seller”, “Buyer”
Amount	decimal	Amount of payment
NameOnAccount	string	Name on account
BankName	string	Bank name
RoutingNumber	string	Routing number
AccountNumber	string	Account number
IsInternational	bool	**(Optional)**
InternationalRC	string	**(Optional)**

Possible HTTP Responses to /api/v1/transaction:

HTTP Status Code	Description

OK (200)	The expected result of all api calls. This indicates that everything worked as expected. The BODY of the http response in this case simply contains a unique key corresponding to transaction that was just created.

Bad Request (400)	This response is used anytime there is validation errors in the request. The response body will include a json object identifying the field names that caused the validation errors and a message indicating what is wrong.
	
	Example response:
	`[`
	`{ "field":"request.Name",`
	`"message":"The Name property is required."`
	`},`
	`{ "field":"request.Amount",`
	`"message":"The Amount property is required."`
	`}`
	`]`
	
	Additionally, if a call attempts to create a transaction in which the buyer, seller, or creator resides in a state where PaySAFE has restricted activity (see note in Service Overview), the response will be:
	
	`"Service is not available in <State>" (where <State> is the state in which the buyer, seller, or creator resides.)`

Server Error (500)	This response indicates that the PaySAFE API cannot recover from an error.

PUT /api/v1/transaction/{transaction_key}/cancel

A PUT to the *transaction/cancel* endpoint cancels or disputes a new transaction. If the transaction has not yet reached the FundsHeld status (35), the transaction will be canceled, and an InEscrow of 0 will be returned in the response. If the status is greater or equal to the FundsHeld status, the transaction will be disputed, and the InEscrow Amount will be sent in the response. Emails will be sent to the buyer and seller informing them of the cancellation or dispute.

Property	Description
Key	Optional since the transaction key is already in the URL
Comment	Optional. Used to create a log entry stating the reason for the cancel request.

Possible HTTP Responses to /api/v1/transaction/cancel:

HTTP Status Code	Description
OK (200)	The expected result of all api calls. This indicates that everything worked as expected. The BODY of the http response in this case simply contains a unique key corresponding to transaction that was just created.

Bad Request (400)	This response is used anytime there is validation errors in the request. The response body will include a json object identifying the field names that caused the validation errors and a message indicating what is wrong.
	
	Example response:
	`[`
	`{ "field":"request.Name",`
	`"message":"The Name property is required."`
	`},`
	`{ "field":"request.Amount",`
	`"message":"The Amount property is required."`
	`}`
	`]`

Server Error (500)	This response indicates that the PaySAFE API cannot recover from an error.
Response Body	1. InEscrow 2. TransactionKey 3. CancelDate

POST/api/v1/InspectionPeriod/change

A POST to the *InspectionPeriod/change* endpoint will updated the transaction's inspection hours to the supplied value. It will return an error if the transaction is past the Payment Approved status stage (70).

***InspectionPeriod is the number of hours to hold the payment schedule within PaymentRequested status. All changes to InspectionPeriod are added to the original date the transaction reached the PaymentRequested status. (E.g. an InspectionPeriod change replaces the original Inspection period. It DOES NOT add the new hours to the original hours.)**

Property	Description
Key	The Transaction Key
NewInspectionHours	The integer value of the new Inspection hours.

GET /api/v1/feeschedule/get

A GET to the feeschedule endpoint will get the fee schedule for the partner code associated with the basic authentication parameter**.**

POST /api/v1/paymentchedule/{transaction_key}/ExecuteSchedule

This POST will execute the current payment schedule on the provided transaction.

Property	Description
Key	The key to the transaction
Note	Optional Note

PUT /api/v1/paymentschedule/Dispute

This API call will take the provided payment schedule and apply it to the transaction based on the passed in key.

Property	Description
TransactionKey	The key to the transaction
List of Payment Items (see object definition of Payment Item below)	A list of all payments for this transaction

Possible HTTP Responses to /api/v1/paymentschedule calls:

HTTP Status Code	Description
OK (200)	The expected result of all api calls. This indicates that everything worked as expected. The BODY of the http response in this case simply contains a unique key corresponding to transaction that was just created.

Bad Request (400)	This response is used anytime there is validation errors in the request. The response body will include a json object identifying the field names that caused the validation errors and a message indicating what is wrong.
	Example response:
	`[`
	`{ "field":"request.Name",`
	`"message":"The Name property is required."`
	`},`
	`{ "field":"request.Amount",`
	`"message":"The Amount property is required."`
	`}`
	`]`

Server Error (500)	This response indicates that the PaySAFE API cannot recover from an error.
Response Body	`[` `{ "TransactionKey":"12345678",`

	"InEscrow":"4321.12" }]

Transaction Notifications

PaySAFE has implemented a transaction notification service to communicate changes that occur to transactions created by our partners through the API. Rather than having an endpoint that needs to be polled for status changes by the integrator, PaySAFE instead publishes changes in real time to the CallbackUrl endpoint specified in the create transaction request.

The BODY of the HTTP POST is a simple JSON object that has the following structure:

```
{
      "ExternalId": "ABC123",
      "Status": "Active",
      "StatusId": "40",
      "PaySafeFeeRealized": 60.00,
      "PaySafeFee": 60.00,
      "TotalDeposits": 400.00,
      "TotalPayments": 0.00,
      "ContractValue": 400.00,
      "InEscrow": 340.00,
      "EventDate": "2014-01-28T12:00:18Z",
      "Environment": "Test"
}
```

ExternalId is the value provided in the original create request.

Status is a string representation of the status a contract transitioned into. Currently only the subset of status' below are published. A complete list of all the possible status' transactions can move to/from are included at the end of this document in the Appendix.
The current list of published events are as follows:

- Estimate
- PendingApproval
- EstimateRejected
- EstimateApproved
- FundsHeld
- Active
- PaymentRequested
- RefundRequested
- RefundRejected
- PaymentApproved
- RefundApproved
- Closed
- Cancelled
- Dispute

StatusID is the integer version of the status (see reference at the end of the document for translations).

PaySafeFee is the PaySAFE escrow fee amount allocated to the transaction

PaySafeFeeRealized is the total amount applied towards the PaySafeFee

TotalDeposits is the total amount of all deposits received into escrow for the transaction

TotalPayments is the total amount of all payments disbursed

ContractValue equals the total amount of the original escrow contract

InEscrow is the total amount held in Escrow. (Amount of payments received - Escrow fee)

EventDate is the UTC time-stamp of when the event fired.

Environment indicates the source of the event. Possible options are: [Local|Dev|Test|Prod]

Notification for deposits applied to a transaction

This will be in the response body of any deposits applied to the transaction:


```
{
      "Key": "55435245235",
      "NotificationType": "PaymentDeposit",
      "Status": "FundsHeld",
      "StatusId": "40",
      "ExternalId": "ABC123",
      "PaySafeFeeRealized": 60.00,
      "PaySafeFee": 60.00,
      "DepositAmount": 150.00,
      "TotalDeposits": 400.00,
      "TotalPayments": 0.00
      "ContractValue": 400.00,
      "InEscrow": 340.00,
      "EventDate": "2014-01-28T12:00:18Z",
      "Environment": "Test"
}
```

Appendix

Transaction Status Reference:

The transaction status reference below gives a good overview of how a transaction moves from the original estimate state to closed.

(0) Estimate: This status indicates that a transaction is still being created. Transactions in this status can only be seen by the creator. Some validation is not necessary to save an estimate. Only database constraints are validated for transactions in this status.

(10) PendingApproval: Once a transaction creator has "Invited" the other members, a transaction is placed into Pending Approval Status. At this time, all members of the transaction can view the transaction. (Visible in the Overview page) Any non-creator members should be given an opportunity to "Accept" the estimate at this status.

(20) EstimateRejected: When at least one member (buyer or seller) rejects the estimate, this is the status that a transaction is moved to. At this point only the creator can interact with the transaction. Any "Updates" to the transaction set the status back to "Estimate". (at this point the whole approval flow can resume)

(30) EstimateApproved: As soon as both the buyer and the seller approve the transaction, the status is moved to this state. In this state no member can do anything to the transaction itself. The transaction is waiting for the buyer to fund the transaction. Note that comments and attachments can be added at any time. An Admin task is required to move the status forward.

(35) FundsHeld: This state is reached after an admin has confirmed that a deposit has been made, but the hold period has not yet transpired. Once an Admin "Clears" the deposit, the transaction status can move to Active.

(40) Active: Once a PaySAFE rep has confirmation that funds have hit the escrow deposit account, and that the internal hold period has expired, the admin can progress the transaction to the Active state. This indicates to the buyer and seller that the funds are guaranteed and the transaction can commence.

(50) PaymentRequested: A transaction is placed into this status when the buyer requests a partial or full/final payment. Typically the buyer will provide some type of communication to the seller if the payment is not for the complete contract value. Either way, in this status it's up to the seller to progress the transaction by accepting or rejecting the payment. In a broker transaction the payment request status is set by the buyer and indicates that they are requesting that the payment schedule defined and agreed to by all members when the transaction was first created and approved.

(55) RefundRequested: A Refund Requested by the buyer triggers this status. The same business rules apply as with PaymentRequest (50). The only difference is who the recipient of the funds are and who needs to approve the request (seller).

(60) PaymentRejected: The buyer triggers this status when they do not agree to the payment. The buyer has the option to add more documentation and resubmit a payment request. It is possible that the transaction stays in this status an an arbitration is required to settle the dispute. The transaction can go in two directions here, the seller can re-submit the payment request, or an admin sets the status manually to dispute, and the money in escrow is disbursed by court order outside of the system (for now)

(65) RefundRejected: Triggered when the Seller rejects a buyer's refund request.

(70) PaymentApproved: This status is achieved when the seller accepts the payment request defined by the buyer, be it partial, full/final. In the case of a broker transaction, the seller accepts the execution of the payment schedule.

(75) RefundApproved: Buyer Refund Approved

(80) Closed: This status is set when a PaySAFE rep releases the funds (does an ACH transfer) matching the payment terms. In the case of a broker flow, when the LAST outstanding payment is "released", the transaction status moves to Closed. At this point the transaction is finished. There is nothing left to do.

(90) Cancelled: A transaction can be cancelled anytime up until a deposit is made. Only the creator can cancel a transaction.

(100) Dispute: If there is no agreement between the buyer and seller about how to release the funds held in escrow, a PaySAFE rep will set the transaction to this status.

Sample Transaction API payload:

{"Key":null,
"Type":1,
"Status":30,
"Name":"Test Transaction",
"PromoCode":null,
"TrackingCode":null,
"Description":"Description of test transaction",
"TemplateData":null,
"ReleaseTerms":"On ship notification",
"ClassifiedAdUrl":null,
"CreatorId":"-4877232327969916574",
"BuyerId":"-5540123942226871818",
"SellerId":"-2599362665932899381",
"SalePrice":5000.0,
"SupressEmails":true,
"CallbackUrl":null,
"ExternalId":"ProxyInternalCode",
"InspectionHours":0,
"AutoRequestPayment":true,
"Origin":"api",
"PartnerCode":"quest",
“AutoApprovePayment”:true,
"IsBrokered":true,
"Payments":[{"Label":"QuestBank",
"Member":"Seller",
"Amount":2000.0,
"NameOnAccount":"QuestAccount",
"BankName":"Quest Banking",

"RoutingNumber":"044015747",

"AccountNumber":"123",

"IsInternational":false,

"InternationalRC":null},

{"Label":"Zin Bank",

"Member":"Creator",

"Amount":2901.0, "NameOnAccount":"Zin Account",

"BankName":"Zin Banking",

"RoutingNumber":"044015747",

"AccountNumber":"124",

"IsInternational":false,

"InternationalRC":null}]}