ReceivablesPro API Documentation

To send messages to the Receivables Pro API, users must obtain a UserName and SecretAccessKey from the Payliance Integration team.

Requests are written in JSON and all examples within this document are presented in JSON format.

Sandbox testing is required before sending live API transactions to the gateway for processing. Before receiving credentials for the Live API, Payliance requires the merchant demonstrates the following capabilities in the sandbox:

- Create a new Customer Object
- Create a new ACH Object
- Create new Payment Objects
    - One-time Payment
    - Recurring Schedule

Once the Payliance team certifies the above criteria has been met, the Live credentials will be provided. Test data sent to the sandbox **will not** be sent out to banking partners for processing.

API Endpoints:

Staging: https://staging.gatewayapi.payliance.com

Production: https://gatewayapi.payliance.com

Contents

Creating and Managing Customer Objects 4
Creating a new customer object 4
Required Attributes 4
Optional Attributes 4
Sample Request 5
Editing Customer Objects 5
Retrieving information about a specific customer 7
Required Attributes 7
Sample Request 7
Retrieving a complete list of customers 8
Sample Request 8
Pagination 10
Creating and Managing ACH Account Objects 11
Creating a new ACH account object for an existing customer 11
Required Attributes 11
Sample Request 11
Retrieving ACH Accounts for Customers 12
Sample Request 12
Retrieving ACH Account Object Details 12
Sample Request 12
Retrieving the Default ACH Payment Account 13
Sample Request 13
Setting a Default ACH Account for a Customer Object 13
Sample Request 13
Updating ACH Account Objects 14
Sample Request 14
Deleting an ACH Account Object 14
Sample Request 14
Creating and Managing Payment Objects 15
Payment Object Response Attributes 15
One-Time Payment Objects 15
Creating a Payment Object for an Existing Customer 15

Creating a new One-Time Payment Object with a new ACH Account Object for an Existing Customer 17
Creating a One-Time Payment Object for a New Customer 19
Voiding a Payment 20
Sample Request 21
Refunding (Reversing) a Payment 22
Sample Request 22
Recurring Schedule Payment Objects 23
Recurring Payment Object Response Attributes 23
Creating a new Recurring Schedule for an Existing Customer & ACH Object 23
Creating a new Recurring Schedule for an Existing Customer & new ACH Object 25
Creating a new Recurring Schedule Object for a New Customer Object 26
Entering a One-time Payment for A Future Date 29
Editing an Existing Recurring Payment Schedule 29
Suspending a Recurring Payment Schedule 31
Pausing a Recurring Payment Schedule 32
Resuming a Recurring Payment Schedule 33
Deleting a Payment Schedule 34
Payment Plan Objects 35
Payment Plan Object Response Attributes 35
Creating a new Payment Plan Object for an Existing Customer & ACH Object 35
Retrieving Payment Details 37
Single Transaction Query 37
Sample Request 37
Multiple Transaction Query 38
Retrieving Payments for a Recurring Payment Schedule Object 41
Retrieving Payments for a Payment Plan Schedule Object 44
Retrieving Information on All Schedules (Recurring & Payment Plans) 45
Retrieving Payment Schedules Associated with A Customer 47
Payliance Failure Codes 48

Creating and Managing Customer Objects

To process payments and enter schedules you must first create a Customer Record. Once this record is created, you can attach bank accounts to it. When processing payments and schedules, you specify the "Id" for the payment account, which instructs the system to attach the payment/schedule to the associated Customer Record.

Customer Records can be created, viewed, and managed via the API. The API can also be used to view and manage Customer Records created via the web-based UI or via Web Payment Pages.

Creating a new customer object

POST messages to endpoint + /api/ReceivablesProCustomer/CreateCustomer

Required Attributes

- FirstName: String -- 150 characters max
- LastName: String -- 150 characters max
- ShippingSameAsBilling: Boolean -- Set to "true" to indicate that there is no separate shipping address, even if you are not including a billing address. Set to "false" to require ShippingAddress fields.

Optional Attributes

- BillingAddress: Object -- billing address is not required, but if included all its attributes are required except "StreetAddress2," "StateCode," and "Country." Up to 10 characters of any type is valid for "ZipCode."
  - StreetAddress1: String -- 250 characters max
  - StreetAddress2: String -- optional, 250 characters max
  - City: String -- 100 characters max
  - StateCode: String -- valid 2-character State or Territory code in the US or Canada
  - ZipCode: String -- up to 10 characters
  - Country: String -- optional, defaults to "US," max 3 characters.
- ShippingAddress: Object -- not required if ShippingSameAsBilling is set to "true," but if included all its attributes are required except "StreetAddress2," "StateCode," and "Country." Up to 10 characters of any type is valid for "ZipCode."
  - StreetAddress1: String -- 250 characters max
  - StreetAddress2: String -- optional, 250 characters max
  - City: String -- 100 characters max
  - StateCode: String -- valid 2-character State or Territory code in the US or Canada
  - ZipCode: String -- up to 10 characters
  - Country: String -- optional, defaults to "US," max 3 characters.
- Company: String -- 50 characters max
- CustomerAccount: String -- 28 characters max. This is a non-system account number you can assign.
- Phone: String -- 10 characters max, enter as a numeric string; do not include parenthesis or dashes.
- AltPhone: String -- 10 characters max, enter as a numeric string; do not include parenthesis or dashes.
- MobilePhone: String -- 10 characters max, enter as a numeric string; do not include parenthesis or dashes.
- Fax: String -- 10 characters max, enter as a numeric string; do not include parenthesis or dashes.
- Email: String -- 100 characters max, must be a valid email address.
- AltEmail: String -- 100 characters max, must be a valid email address.
- Website: String -- 100 characters max, must be in URL format.
- Notes: String -- 2048 characters max; an open text field.

Sample Request

Request:

```
{
 "Request": {
  "FirstName": "Patrick",
  "LastName": "Exeter",
  "ShippingSameAsBilling": true,
 },
 "Auth": {
  "UserName": "APIUserxxx724",
  "SecretAccessKey": "Secret Access Key"
 }
}
```

Response:

```
{
  "Response": {
    "Id": 390839,
    "CustomerAccount": null,
    "FirstName": "Patrick",
    "MiddleName": null,
    "LastName": "Exeter",
    "BillingAddress": null,
    "ShippingSameAsBilling": true,
    "ShippingAddress": null,
    "Company": null,
    "Phone": null,
    "AltPhone": null,
    "MobilePhone": null,
    "Fax": null,
    "Email": null,
    "AltEmail": null,
    "Website": null,
    "Notes": null,
    "CreatedOn": "2017-06-23T19:04:34.500093Z",
    "LastModified": "2017-06-23T19:04:34.500093Z"
  },
  "Success": true,
  "Message": null
}
```

The Id in the response will be associated with this customer object in future requests.

Editing Customer Objects

POST messages to endpoint + /api/ReceivablesProCustomer/UpdateCustomer

You can use the API to make changes to the following Customer Object Attributes

- AltEmail: String -- can be changed to any valid email address or set to null.
- AltPhone: String -- can be changed to any other 10 digit string or set to null.
- MobilePhone: String -- can be changed to any other 10 digit string or set to null.
- Fax: String -- can be changed to any other 10 digit string or set to null.
- Website: String -- 100 characters max, must be in URL format. Can be changed to any other URL or set to null.
- BillingAddress: Object -- billing address is not required, but if included all its attributes are required except "StreetAddress2," "StateCode," and "Country." Up to 10 characters of any type is valid for "ZipCode."
    - StreetAddress1: String -- first line of billing street address can be changed as required.
    - StreetAddress2: String -- second line of billing street address can be changed as required or set to null.
    - City: String -- can be changed as required.
    - StateCode: String -- valid 2-character State or Territory code in the US or Canada can be changed as required.
    - ZipCode: String -- billing postal code, up to 10 characters. Can be changed as required.
    - Country: String -- defaults to "US," max 3 characters. Can be changed to "US" or "CAN."
- ShippingSameAsBilling: Boolean -- Set to "true" to indicate that there is no separate shipping address, even if you are not including a billing address. If "true" the system will set shipping field values equal to their corresponding billing address fields. If set to "false" ShippingAddress fields are not linked to billing address fields and may be entered separately.
- ShippingAddress: Object -- shipping address fields for the customer record. If ShippingSameAsBilling is "true" these fields are linked to their corresponding billing fields and cannot be individually changed. Otherwise the

values are independent of billing address and may be changed or set to null. If one Shipping Address field is used, then all are required except StreetAddress2, StateCode, and Country.

  - StreetAddress1: String -- primary shipping address line can be changed as required.
  - StreetAddress2: String -- second shipping address line can be changed as required or set to null.
  - City: String -- can be changed as required.
  - StateCode: String -- valid 2-character State or Territory code in the US or Canada can be changed as required.
  - ZipCode: String -- the shipping postal-code, up to 10 characters. Can be changed as required.
  - Country: String -- defaults to "US," max 3 characters. Can be changed to "US" or "CAN."
- Company: String -- the Company name for the customer can be changed as desired.
- Notes: String -- an open descriptive field can be changed as required.
- CustomerAccount: String -- the Account Number you assigned to the Customer. Can be changed as required.
- FirstName: String -- the customer's last name can be changed as required.
- LastName: String -- the customer's first name can be changed as required.
- Email: String -- 100 characters max, must be a valid email address. Can be changed as required.
- Phone: String -- 10 characters max, a numeric string without parenthesis or dashes. Can be changed as required.

NOTE: If changes are entered for any other fields, the system will ignore them and retain existing values. You must include the entire set of Customer Object Attributes in your change request, even if you are changing just one of the fields. Failure to do this will result in any non-included fields being set to null.
NOTE: If you know a field is set to null, and you want to keep it that way, you do not need to include it in the Request.

Sample Request

Request

```
{
 "Request": {
"Id": 390927,
          "CustomerAccount": null,
          "FirstName": "Axil",
          "MiddleName": null,
          "LastName": "Mason",
            "BillingAddress": {
            "StreetAddress1": "1234 Main St.",
            "City": "Columbus",
            "StateCode": "OH",
            "ZipCode": "43215",
  },
         "ShippingSameAsBilling": true,
          "ShippingAddress": null,
          "Company": "Axil's Axles",
          "Phone": null,
          "AltPhone": null,
          "MobilePhone": null,
          "Fax": null,
          "Email": null,
          "AltEmail": null,
          "Website": null,
          "Notes": null,
          "CreatedOn": "2017-06-25T17:29:58Z",
          "LastModified": "2017-06-25T17:29:58Z"
},
 "Auth": {
 "UserName": "APIUser***724",
   "SecretAccessKey": "Secret Access Key"
 }
}
```

Response:

```
{
  "Response": {
    "Id": 390927,
    "CustomerAccount": null,
    "FirstName": "Axil",
    "MiddleName": null,
    "LastName": "Mason",
    "BillingAddress": {
      "StreetAddress1": "1234 Main St.",
      "StreetAddress2": null,
      "City": "Columbus",
      "StateCode": 43,
      "ZipCode": "43215",
      "Country": 0
    },
    "ShippingSameAsBilling": true,
    "ShippingAddress": {
      "StreetAddress1": "1234 Main St.",
      "StreetAddress2": null,
      "City": "Columbus",
      "StateCode": 43,
      "ZipCode": "43215",
      "Country": 0
    },
    "Company": "Axil's Axles",
    "Phone": null,
    "AltPhone": null,
    "MobilePhone": null,
    "Fax": null,
    "Email": null,
    "AltEmail": null,
    "Website": null,
    "Notes": null,
    "CreatedOn": "2017-06-25T17:29:58Z",
    "LastModified": "2017-06-25T19:58:24.3855756Z"
  },
  "Success": true,
  "Message": null
```

Retrieving information about a specific customer

POST messages to endpoint + /api/ReceivablesProCustomer/Customer

Required Attributes

CustomerID to be inserted into the “Request” field.

Sample Request

Request:

```
{
 "Request": 390839,
 "Auth": {
  "UserName": "APIUserxxx724",
  "SecretAccessKey": "Secret Access Key"
 }
}
```

Response:

```
{
  "Response": {
    "Id": 390839,
    "CustomerAccount": null,
    "FirstName": "Patrick",
    "MiddleName": null,
    "LastName": "Exeter",
    "BillingAddress": null,
    "ShippingSameAsBilling": true,
    "ShippingAddress": null,
    "Company": null,
```

	"Phone": null, "AltPhone": null, "MobilePhone": null, "Fax": null, "Email": null, "AltEmail": null, "Website": null, "Notes": null, "CreatedOn": "2017-06-23T19:04:34Z", "LastModified": "2017-06-23T19:04:34Z" }, "Success": true, "Message": null }

Retrieving a complete list of customers

POST message to endpoint + /api/ReceivablesProCustomer/Customers

The API can be used to return a list of all Customer Records. The generated list can be in “complete” format or “lite” format, which is ideal for creating summary tables and snapshots.

Because the full list of customers can grow quite large and unwieldy, the API uses paging to limit the number of results returned from a single Request. The default is to show the oldest 200 customers. You can use Pagination filters to govern both the number of results returned in the Response as well as the Response page you would like to retrieve.

Sample Request

Request:	Response:

{
"SortBy": 0,
"Direction": 0,
"Page": 1,
"PageSize": 3,
"Lite": true,
"Auth": {
"UserName": "APIUserxxx724",
"SecretAccessKey": "Secret Access Key"
}
}
	{
	"Response": {
	"Page": 1,
	"ItemsPerPage": 3,
	"TotalItems": 185,
	"TotalPages": 62,
	"Items": [
	{
	"Id": 380150,
	"CustomerAccount": null,
	"FirstName": "Aaron",
	"MiddleName": null,
	"LastName": "Jackson",
	"BillingAddress": null,
	"ShippingSameAsBilling": true,
	"ShippingAddress": null,
	"Company": "",
	"Phone": null,
	"AltPhone": null,
	"MobilePhone": null,
	"Fax": null,
	"Email": null,
	"AltEmail": null,
	"Website": null,
	"Notes": null,
	"CreatedOn": null,
	"LastModified": null
	},
	{
	"Id": 380155,
	"CustomerAccount": null,
	"FirstName": "Abbi",
	"MiddleName": null,

```
            "LastName": "Hayward",
            "BillingAddress": null,
            "ShippingSameAsBilling": true,
            "ShippingAddress": null,
            "Company": "",
            "Phone": null,
            "AltPhone": null,
            "MobilePhone": null,
            "Fax": null,
            "Email": null,
            "AltEmail": null,
            "Website": null,
            "Notes": null,
            "CreatedOn": null,
            "LastModified": null
        },
        {
            "Id": 380103,
            "CustomerAccount": null,
            "FirstName": "Ainsley",
            "MiddleName": null,
            "LastName": "Peacock",
            "BillingAddress": null,
            "ShippingSameAsBilling": true,
            "ShippingAddress": null,
            "Company": "",
            "Phone": null,
            "AltPhone": null,
            "MobilePhone": null,
            "Fax": null,
            "Email": null,
            "AltEmail": null,
            "Website": null,
            "Notes": null,
            "CreatedOn": null,
            "LastModified": null
        }
    ]
  },
  "Success": true,
  "Message": null
}
```

Pagination

- SortBy – default sort is ascending by CustomerID number (listing oldest customer first)

Code	Description
0	FirstName
1	MiddleName
2	LastName
3	Company
4	BillingAddressCity
5	BillingAddressState
6	BillingAddressZip
7	BillingAddressCountry
8	ShippingAddressCity
9	ShippingAddressState
10	ShippingAddressZip
11	ShippingAddressCountry

- Direction: 0 = ascending; 1 = descending
- Page = the set of results you want to return. “1” returns the first page, “2” the second, etc. If this filter is not included, the Request defaults to “1.”
- Pagesize = the number of customer records per page. If this filter is not included, the Request defaults to “200.”

Creating and Managing ACH Account Objects

Each customer object will require at least one ACH object associated with it as a default account for payments. The Payment Account Objects are used when creating payments and payment schedules. The Customer Object associated with the Payment Account Object specified in the New Payment or New Schedule Request is associated with the newly created Payment or Schedule Object.

For each Customer Object, one bank account can be set at the default ACH object. Default accounts enable more direct routes for processing payments and entering schedules, as they remove the need to select one of several saved payment accounts prior to processing a payment.

The API can be used to enter new ACH Account Objects, as well as to view saved accounts, edit saved accounts, delete saved accounts, manage default accounts, and list all accounts associated with an individual Customer Object.

Creating a new ACH account object for an existing customer

POST messages to endpoint + /api/ReceivablesProAccount/CreateAchAccount

Required Attributes

- CustomerId: Integer -- the system identifier for the Customer Record to which the card is being added.
- RoutingNumber: String -- 9-digit bank routing number
- AccountNumber: String -- min 4 digits, max 100 digits; the bank account number
- BankName: String-- 100 characters max; the financial institution name or nickname to associate with the bank account
- IsCheckingAccount: Boolean -- enter “true” to indicate a checking account or “false” to indicate a savings account. General Ledger accounts are not supported.
- IsDefault: Boolean -- enter “true” to set the account as the default ACH Account for the Customer Record, otherwise enter “false.” If there are no other ACH Accounts attached to the Customer Record, the field will default to “true” regardless of your entry.

Sample Request

Request:

```
{
 "Request": {
     "AccountNumber": "8774598",
     "RoutingNumber": "011201458",
     "BankName": "Test API Account",
     "IsCheckingAccount": true,
     "CustomerId": 390839,
     "IsDefault": true,

 },
 "Auth": {
  "UserName": " APIUserxxx724",
  "SecretAccessKey": " Secret Access Key"
 }
}
```

Response:

```
{
  "Response": {
     "AccountNumber": "***4598",
     "RoutingNumber": "011201458",
     "BankName": "Test API Account",
     "IsCheckingAccount": true,
     "Id": 585373,
     "CustomerId": 390839,
     "IsDefault": true,
     "CreatedOn": "2017-06-23T22:29:21.2002449Z",
     "LastModified": "2017-06-23T22:29:21.2002449Z"
  },
  "Success": true,
  "Message": null
}
```

The ID generated in the response is the ACH Account ID that can be associated with payment requests using this customer’s ACH account in future requests.

Retrieving ACH Accounts for Customers

POST messages to endpoint + /api/ReceivablesProCustomer/AchAccounts

ReceivablesPro can store multiple ACH Account objects for each Customer object. To retrieve a list of ACH accounts associated with a customer, use the Customer ID to launch the request.

Sample Request

Request:

```
{
 "Request": 390839,
 "Auth": {
 "UserName": "APIUser***724",
  "SecretAccessKey": "Secret Access Key"
 }
}
```

Response:

```
{
  "Response": [
    {
      "AccountNumber": "***4598",
      "RoutingNumber": "011201458",
      "BankName": "Test API Account",
      "IsCheckingAccount": true,
      "Id": 585373,
      "CustomerId": 390839,
      "IsDefault": true,
      "CreatedOn": "2017-06-23T22:29:21Z",
      "LastModified": "2017-06-23T22:29:21Z"
    },
    {
      "AccountNumber": "***8574",
      "RoutingNumber": "011201458",
      "BankName": "Test API Account",
      "IsCheckingAccount": false,
      "Id": 585376,
      "CustomerId": 390839,
      "IsDefault": false,
      "CreatedOn": "2017-06-23T22:35:54Z",
      "LastModified": "2017-06-23T22:35:54Z"
    }
  ],
  "Success": true,
  "Message": null
}
```

Retrieving ACH Account Object Details

POST messages to endpoint + /api/ReceivablesProAccount/AchAccount

To retrieve the details about an ACH Account object, use the ACH Account ID to launch the request.

Sample Request

Request:

```
{
 "Request": 585373,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
  "Response": {
    "AccountNumber": "***4598",
    "RoutingNumber": "011201458",
    "BankName": "Savings Account",
    "IsCheckingAccount": false,
    "Id": 585373,
    "CustomerId": 390839,
    "IsDefault": true,
    "CreatedOn": "2017-06-23T22:29:21Z",
    "LastModified": "2017-06-25T16:13:37Z"
  },
  "Success": true,
  "Message": null
}
```

Retrieving the Default ACH Payment Account

POST messages to endpoint + /api/ReceivablesProCustomer/DefaultAchAccount

For customer objects with more than one associated ACH Account Object, API can return the default account that for a customer.

Sample Request

Request:

```
{
 "Request": 390839,
 "Auth": {
              "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
  "Response": {
    "AccountNumber": "***4598",
    "RoutingNumber": "011201458",
    "BankName": "Savings Account",
    "IsCheckingAccount": false,
    "Id": 585373,
    "CustomerId": 390839,
    "IsDefault": true,
    "CreatedOn": "2017-06-23T22:29:21Z",
    "LastModified": "2017-06-25T16:13:37Z"
  },
  "Success": true,
  "Message": null
}
```

Setting a Default ACH Account for a Customer Object

POST messages to endpoint + /api/ReceivablesProCustomer/SetDefaultAccount

For customer objects with more than one associated ACH Account Object, API can change the default account for the customer object. To submit the request, include the Customer ID and the AccountID of the account you wish to make default.

Sample Request

Request:

```
{
 "CustomerId": 390839,
 "AccountId": 585423,
 "Auth": {
 "UserName": "APIUser154724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
  "Response": true,
  "Success": true,
  "Message": null
}
```

Updating ACH Account Objects

POST messages to endpoint + /api/ReceivablesProAccount/UpdateAchAccount

Only the Account Type (checking/savings) and the Bank Name can be changed. To make any other change to a saved account, you must enter the new account information as a new payment account object and then delete the old payment account object.

Sample Request

Request:

```
{
 "Request": {
        "AccountNumber": "***4598",
        "RoutingNumber": "011201458",
        "BankName": "Savings Account",
        "IsCheckingAccount": false,
        "Id": 585373,
        "CustomerId": 390839,
        "IsDefault": true,
        "CreatedOn": "2017-06-23T22:29:21Z",
        "LastModified": "2017-06-23T22:29:21Z"
          },
 "Auth": {
  "UserName": " APIUserxxx724",
  "SecretAccessKey": "Secret Access Key"
 }
}
```

Response:

```
{
  "Response": {
    "AccountNumber": "***4598",
    "RoutingNumber": "011201458",
    "BankName": "Savings Account",
    "IsCheckingAccount": false,
    "Id": 585373,
    "CustomerId": 390839,
    "IsDefault": true,
    "CreatedOn": "2017-06-23T22:29:21Z",
    "LastModified": "2017-06-25T16:12:24.113518Z"
  },
  "Success": true,
  "Message": null
}
```

Deleting an ACH Account Object

POST messages to endpoint + /api/ReceivablesProAccount/RemoveAchAccount

The API can remove an ACH account from an existing customer record. If the account deleted is listed as the default account, the system will automatically assign the next oldest ACH Account Object associated with the customer as the new default account.

Sample Request

Request:

```
{
 "Request": 585376,
 "Auth": {
 "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
  "Response": true,
  "Success": true,
  "Message": null
}
```

Creating and Managing Payment Objects

The API allows for the creation of three types of Payment Objects.

- One Time Payment – a single transaction to be processed in the next available processing window
- Payment Plans - Schedule configured to discharge a set amount via a set number of payments.
- Recurring Payments - Schedules configured to charge a set amount for a set period, or indefinitely.

Payment Object Response Attributes

- Id: Integer -- this is the PaymentId number you will need to reference in future calls related to this transaction including void, refund, and view details.
- Status

Code	Description
0	Pending
2*	Failed
4*	Posted
5	Refund (settled)
7	Refunded
9	ReversePosted
10	Settled
11	Voided
15	ReverseNSF / Returned NSF

***NOTE**: These labels do not all correspond to Status names in the UI. Mapping is as follows: Reversed = Refunded; ReversePosted = Refund (Posted); ReverseNSF = Returned NSF*

** expected responses for new payment objects*

- TraceNumber: String -- a pass-through field that displays the processor's transaction identifier for a successful transaction or the provider's error message for a failed transaction.
- FailureData: String -- standardized system information about why a transaction failed; 4 components:
  - Code --a 4-digit Error Code (See Payliance Failure Codes for details)
  - Description --a description of the Error
  - MerchantActionText --instructions to the merchant on how to handle the error.
  - IsDecline" – will always display null

One-Time Payment Objects

Creating a Payment Object for an Existing Customer

POST messages to endpoint + /api/ReceivablesProPayment/CreatePayment

Required Attributes

- AccountId: Integer -- the system identifier for the ACH Account used for the payment. This is the "Id" attribute from the ACH Account Object created previously.
- Amount: Number -- the amount to charge. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 3.129 will result in a payment Amount of 3.13.

Sample Request

Request:

```
{
 "Request": {
  "AccountId": 585373,
  "Amount": 25
 },
 "Auth": {
  "UserName": " APIUserxxx724",
  "SecretAccessKey": "Secret Access Key"
 }
}
```

Response:


```
{
  "Response": {
    "Id": 4511405,
    "AccountId": 585373,
    "Amount": 25,
    "IsDebit": false,
    "Cvv": null,
    "PaymentSubType": 5,
    "InvoiceId": null,
    "InvoiceNumber": null,
    "PurchaseOrderNumber": null,
    "OrderId": null,
    "Description": null,
    "Latitude": null,
    "Longitude": null,
    "SuccessReceiptOptions": null,
    "FailureReceiptOptions": null,
    "CustomerId": 390839,
    "CustomerFirstName": "Patrick",
    "CustomerLastName": "Exeter",
    "CustomerCompany": null,
    "ReferenceId": 0,
    "Status": 4,
    "RecurringScheduleId": 0,
    "PaymentType": 2,
    "ProviderAuthCode": "Approved",
    "TraceNumber": "bb270335-d184-436e-a3e8-74bda1219989",
    "PaymentDate": "2017-06-25T06:00:00Z",
    "ReturnDate": null,
    "EstimatedSettleDate": "2017-06-29T06:00:00Z",
    "ActualSettleDate": null,
    "CanVoidUntil": "2017-06-25T21:00:00Z",
    "FailureData": null,
    "IsDecline": false,
    "CreatedOn": "2017-06-25T16:50:47Z",
    "LastModified": "2017-06-25T16:50:47Z",
    "RequiresReceipt": false,
    "PaymentToken": null
  },
  "Success": true,
  "Message": null
}
```


Optional Attributes

- IsDebit: Boolean-- this field defaults to “false” to indicate a payment if it is included. Entering “true” will result in an error, as standalone credits are not currently supported via the API.
- PaymentSubType: String-- this field defaults to “WEB” for ACH payments if not included or left null.

Code	Description
0	CCD (Company name must be associated with the Customer Object)
2	PPD
4	TEL
5	WEB

- InvoiceNumber: String-- 50 characters max; the Invoice Number attached to the payment. If the number provided is a system Invoice Number, the associated InvoiceId will be attached to the payment and the payment will be applied to the system invoice.

- PurchaseOrderNumber: String-- 50 characters max; the Purchase Order Number attached to the payment.
- OrderId: String-- 50 characters max; a back-end transaction identifier not included in the UI.
- Description: String-- -- 2048 characters max.
- Latitude: Number-- this field is not currently used in the UI, but will accept and save latitude values.
- Longitude: Number-- this field is not currently used in the UI, but will accept and save longitude values.
- SuccessReceiptOptions: Object -- the receipt instructions. Null means the current system default settings should be used. (If not included, null is the default.) The following additional fields can be used to override defaults and direct how payment receipts are sent:
- SendToCustomer --"true" indicates that the receipt gets sent to the email address attached to the Customer Record; "false" indicates that it does not.
- SendToOtherAddresses-- a specific email address (or addresses) to which the receipt email is to be sent. Using brackets, enter valid email addresses separated by commas.
- FailureReceiptOptions: Object -- the failure notification email instructions. Null means the current system default settings should be used. (If not included, null is the default.) The following additional fields can be used to override defaults and direct how failure notifications are sent:
- SendToCustomer --"true" indicates that the notification gets sent to the email address attached to the Customer Record; "false" indicates that it does not.
- SendToOtherAddresses-- a specific email address (or addresses) to which the failure notification email is to be sent. Using brackets, enter valid email addresses separated by commas.

Creating a new One-Time Payment Object with a new ACH Account Object for an Existing Customer

POST messages to endpoint + /api/ReceivablesProPayment/CreateNewAccountPayment

Required Attributes (ACH Account Object)

- CustomerId: Integer -- the system identifier for the Customer Record to which the card is being added. This is the "Id" attribute from the Customer Object created in the previous step.
- RoutingNumber: String -- 9-digit bank routing number
- AccountNumber: String -- min 4 digits, max 100 digits; the bank account number
- BankName: String-- 100 characters max; the financial institution name
- IsCheckingAccount: Boolean -- enter "true" to indicate a checking account or "false" to indicate a savings account. General Ledger accounts are not supported.
- IsDefault: Boolean -- enter "true" to set the account as the default ACH Account for the Customer Record, otherwise enter "false." If there are no other ACH Accounts attached to the Customer Record, the field will default to "true" regardless of your entry.

Required Attributes (Payment Object)

- Amount: Number -- the amount to charge. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 3.129 will result in a payment Amount of 3.13.

Sample Request

Request:

```
{
 "Request": {
  "Account": {
   "AccountNumber": "123456789",
   "RoutingNumber": "122100024",
   "BankName": "API Test",
   "IsCheckingAccount": true,
   "CustomerId": "390926",
   "IsDefault": false

  },
  "Payment": {
   "Amount": 15.0,
   "IsDebit": false
  }
 },
  "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "Secret Access Key"
 }
}
```

Response:

```
{
  "Response": {
    "Account": {
      "AccountNumber": "*****6789",
      "RoutingNumber": "122100024",
      "BankName": "API Test",
      "IsCheckingAccount": true,
      "Id": 585419,
      "CustomerId": 390926,
      "IsDefault": false,
      "CreatedOn": "2017-06-25T18:14:02.9432772Z",
      "LastModified": "2017-06-25T18:14:02.9432772Z"
    },
    "Payment": {
      "Id": 4511409,
      "AccountId": 585419,
      "Amount": 15,
      "IsDebit": false,
      "Cvv": null,
      "PaymentSubType": 5,
      "InvoiceId": null,
      "InvoiceNumber": null,
      "PurchaseOrderNumber": null,
      "OrderId": null,
      "Description": null,
      "Latitude": null,
      "Longitude": null,
      "SuccessReceiptOptions": null,
      "FailureReceiptOptions": null,
      "CustomerId": 390926,
      "CustomerFirstName": "Mildred",
      "CustomerLastName": "Thompson",
      "CustomerCompany": null,
      "ReferenceId": 0,
      "Status": 4,
      "RecurringScheduleId": 0,
      "PaymentType": 2,
      "ProviderAuthCode": "Approved",
      "TraceNumber": "67c7eb32-6049-47bc-9735-b028c541ee59",
      "PaymentDate": "2017-06-25T06:00:00Z",
      "ReturnDate": null,
      "EstimatedSettleDate": "2017-06-29T06:00:00Z",
      "ActualSettleDate": null,
      "CanVoidUntil": "2017-06-25T21:00:00Z",
      "FailureData": null,
      "IsDecline": false,
      "CreatedOn": "2017-06-25T18:14:02Z",
      "LastModified": "2017-06-25T18:14:02Z",
      "RequiresReceipt": false,
      "PaymentToken": null
    }
  },
  "Success": true,
  "Message": null
}
```

The ID assigned in the Payment section of the response is the transaction ID that must be referenced in subsequent actions or queries.

Creating a One-Time Payment Object for a New Customer

POST messages to endpoint + /api/ReceivablesProPayment/CreateNewCustomerPayment

This call combines the creation of a Customer Object, ACH Account Object and Payment Object into one request through the API.

Required Attributes (Customer Object)

- FirstName: String -- 150 characters max
- LastName: String -- 150 characters max
- ShippingSameAsBilling: Boolean -- Set to "true" to indicate that there is no separate shipping address, even if you are not including a billing address. Set to "false" to require ShippingAddress fields.

Required Attributes (ACH Account Object)

- CustomerId: Integer -- the system identifier for the Customer Record to which the card is being added. This is the "Id" attribute from the Customer Object created in the previous step.
- RoutingNumber: String -- 9-digit bank routing number
- AccountNumber: String -- min 4 digits, max 100 digits; the bank account number
- BankName: String-- 100 characters max; the financial institution name
- IsCheckingAccount: Boolean -- enter "true" to indicate a checking account or "false" to indicate a savings account. General Ledger accounts are not supported.
- IsDefault: Boolean -- enter "true" to set the account as the default ACH Account for the Customer Record, otherwise enter "false." If there are no other ACH Accounts attached to the Customer Record, the field will default to "true" regardless of your entry.

Required Attributes (Payment Object)

- Amount: Number -- the amount to charge. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 3.129 will result in a payment Amount of 3.13.

Sample Request

Request:

```
{
 "Request": {
  "Customer": {
   "FirstName": "Mildred",
   "LastName": "Thompson",
   "ShippingSameAsBilling": true
  },
  "Account": {
   "AccountNumber": "985542",
   "RoutingNumber": "011201380",
   "BankName": "CAMDEN NATIONAL BANK",
   "IsCheckingAccount": true,
  },
  "Payment": {
   "Amount": 50,
   "IsDebit": false
  }
 },
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretKey"
 }
}
```

Response:

```
{
   "Response": {
      "Customer": {
         "Id": 390926,
         "CustomerAccount": null,
         "FirstName": "Mildred",
         "MiddleName": null,
         "LastName": "Thompson",
         "BillingAddress": null,
         "ShippingSameAsBilling": true,
         "ShippingAddress": null,
         "Company": null,
         "Phone": null,
         "AltPhone": null,
         "MobilePhone": null,
         "Fax": null,
         "Email": null,
         "AltEmail": null,
         "Website": null,
         "Notes": null,
         "CreatedOn": "2017-06-25T17:26:58.1981555Z",
         "LastModified": "2017-06-25T17:26:58.1981555Z"
      },
      "Account": {
```

```
      "AccountNumber": "**5542",
      "RoutingNumber": "011201380",
      "BankName": "CAMDEN NATIONAL BANK",
      "IsCheckingAccount": true,
      "Id": 585416,
      "CustomerId": 390926,
      "IsDefault": true,
      "CreatedOn": "2017-06-25T17:26:58.6764107Z",
      "LastModified": "2017-06-25T17:26:58.6764107Z"
    },
    "Payment": {
      "Id": 4511406,
      "AccountId": 585416,
      "Amount": 50,
      "IsDebit": false,
      "Cvv": null,
      "PaymentSubType": 5,
      "InvoiceId": null,
      "InvoiceNumber": null,
      "PurchaseOrderNumber": null,
      "OrderId": null,
      "Description": null,
      "Latitude": null,
      "Longitude": null,
      "SuccessReceiptOptions": null,
      "FailureReceiptOptions": null,
      "CustomerId": 390926,
      "CustomerFirstName": "Mildred",
      "CustomerLastName": "Thompson",
      "CustomerCompany": null,
      "ReferenceId": 0,
      "Status": 4,
      "RecurringScheduleId": 0,
      "PaymentType": 2,
      "ProviderAuthCode": "Approved",
      "TraceNumber": "bee11c97-ccce-4012-8f65-de43c876e86f",
      "PaymentDate": "2017-06-25T06:00:00Z",
      "ReturnDate": null,
      "EstimatedSettleDate": "2017-06-29T06:00:00Z",
      "ActualSettleDate": null,
      "CanVoidUntil": "2017-06-25T21:00:00Z",
      "FailureData": null,
      "IsDecline": false,
      "CreatedOn": "2017-06-25T17:26:58Z",
      "LastModified": "2017-06-25T17:26:58Z",
      "RequiresReceipt": false,
      "PaymentToken": null
    }
  },
  "Success": true,
  "Message": null
}
```


Optional Attributes

You may include any of the optional attributes associated with the Customer and Payment Objects in your request.

Voiding a Payment

POST messages to endpoint + /api/ReceivablesProPayment/VoidPayment

Transactions in the following status state are eligible for voids:

- Posted -- Only prior to the ACH batch cut-off time.

- ReversePosted -- The credit transaction generated when a settled ACH transaction is refunded. (Note that t his status displays as “Refund (Posted)” in the UI.)

The “CanVoidUntil” Attribute in any Payment Object provides the date until which the transaction can be voided. *NOTE: The “CanVoidUntil” field is not included in the “lite” record response.*

Voids can only be performed for those transactions that have not been sent into the ACH network yet. transactions can typically be voided until 8:30 PM Eastern Time on the “CanVoidUntil” date. Successfully voided transactions will have their status updated to 11 by the system.

To void a transaction, enter the ID of the transaction to be voided.

Sample Request

Request:

```
{{
 "Request": 4511406,
  "Auth": {
   "UserName": "APIUser***724",
   "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
  "Response": {
    "Id": 4511406,
    "AccountId": 585416,
    "Amount": 50,
    "IsDebit": false,
    "Cvv": null,
    "PaymentSubType": 5,
    "InvoiceId": null,
    "InvoiceNumber": null,
    "PurchaseOrderNumber": null,
    "OrderId": null,
    "Description": null,
    "Latitude": null,
    "Longitude": null,
    "SuccessReceiptOptions": null,
    "FailureReceiptOptions": null,
    "CustomerId": 390926,
    "CustomerFirstName": "Mildred",
    "CustomerLastName": "Thompson",
    "CustomerCompany": null,
    "ReferenceId": 0,
    "Status": 11,
    "RecurringScheduleId": 0,
    "PaymentType": 2,
    "ProviderAuthCode": "Approved",
    "TraceNumber": "bee11c97-ccce-4012-8f65-de43c876e86f",
    "PaymentDate": "2017-06-25T06:00:00Z",
    "ReturnDate": null,
    "EstimatedSettleDate": "2017-06-29T06:00:00Z",
    "ActualSettleDate": null,
    "CanVoidUntil": "2017-06-25T21:00:00Z",
    "FailureData": null,
    "IsDecline": false,
    "CreatedOn": "2017-06-25T17:26:58Z",
    "LastModified": "2017-06-25T18:26:54Z",
    "RequiresReceipt": false,
    "PaymentToken": null
  },
  "Success": true,
  "Message": null
}
```

Refunding (Reversing) a Payment

POST messages to endpoint + /api/ReceivablesProPayment/ReversePayment

Any Settled payment can be refunded via the API. Using the refund call causes the system to perform the following actions:

- Change the Status of the original transaction to “Reversed” (shown as “Refunded” in the UI).
- Create a new credit (negative) transaction with a Status of “ReversePosted” (shown as “Refund (Posted)” in the UI.)
- Set the value of the ReferenceId field for the original transaction to the PaymentId of the credit transaction.
- Set the value of the ReferenceID field for the credit transaction to the PaymentId of the original transaction.

To refund a settled transaction, enter the ID of the transaction to be refunded.

Sample Request

Request:


```
{
 "Request": 4505629,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```


Response:


```
{
  "Response": {
    "Id": 4511411,
    "AccountId": 581242,
    "Amount": 100,
    "IsDebit": true,
    "Cvv": null,
    "PaymentSubType": 2,
    "InvoiceId": null,
    "InvoiceNumber": "",
    "PurchaseOrderNumber": "",
    "OrderId": null,
    "Description": "",
    "Latitude": null,
    "Longitude": null,
    "SuccessReceiptOptions": null,
    "FailureReceiptOptions": null,
    "CustomerId": 386124,
    "CustomerFirstName": "Rex",
    "CustomerLastName": "Winchester",
    "CustomerCompany": null,
    "ReferenceId": 4505629,
    "Status": 9,
    "RecurringScheduleId": 0,
    "PaymentType": 2,
    "ProviderAuthCode": "Approved",
    "TraceNumber": "64379ade-f4c2-4f4a-a31e-e280d4e75328",
    "PaymentDate": "2017-06-25T06:00:00Z",
    "ReturnDate": null,
    "EstimatedSettleDate": "2017-06-27T06:00:00Z",
    "ActualSettleDate": null,
    "CanVoidUntil": "2017-06-25T21:00:00Z",
    "FailureData": null,
    "IsDecline": false,
    "CreatedOn": "2017-06-25T18:30:31Z",
    "LastModified": "2017-06-25T18:30:31Z",
    "RequiresReceipt": false,
    "PaymentToken": null
  },
  "Success": true,
  "Message": null
}
```

Recurring Schedule Payment Objects

If you know the AccountId for the bank account you want to charge, entering a recurring payment schedule via the API can be as simple as entering the payment amount, start date, and frequency along with that AccountId.

Recurring Payment Object Response Attributes

- Id: Integer -- this is the ScheduleId number you will need to reference in future calls related to this schedule including pause, suspend, resume, and view details.
- ScheduleStatus: Integer -- schedule Status set by the system.
  Enumeration: 1 = Active; 3 = Expired; 4 = Suspended
  Expected Response for new schedule is "Active."
- PaymentAmount: Number-- the amount of each scheduled payment, as specified in the Recurring Payment Request. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 23.129 will result in a PaymentAmount of 23.13.
- StartDate: String -- the date on which the schedule will start running, not necessarily the date it will generate its first payment, as specified in the Recurring Payment Request. Date format YYYY-MM-DDT00:00:00Z. (You can ignore the time portion of this date.)
- NOTE: To make schedule management easier, it is strongly recommended that you set the schedule StartDate to the first day it will generate a payment.
- NextScheduleDate: String -- the next date on which the schedule will generate a payment, based on the schedule frequency. Date format YYYY-MM-DDT00:00:00Z. (You can ignore the time portion of this date.)
  NOTE: If you configured the schedule to start on the current day the StartDate will be the current date and the NextScheduleDate will be the future date of the second scheduled payment. If the schedule was configured to start on the future date of its first scheduled payment, StartDate and NextSchduleDate will be the same future date.
- The above assumes you are starting with a valid AccountId, which assumes an existing Customer Object with existing Payment Account Objects attached to it. In many instances, this will not be the case. The following sections cover the complete process for entering recurring payment schedules for completely new customers, for existing customers with existing payment accounts, and for existing customers with new payment accounts.

Creating a new Recurring Schedule for an Existing Customer & ACH Object

POST messages to endpoint + /api/ReceivablesProPaymentSchedule/CreateRecurringPayment

Required Attributes

- AccountId: Integer -- the system identifier for the ACH Account used for the schedule. This is the "Id" attribute from the ACH Account Object created in the previously. The Customer Object associated with the AccountId will be associated with the schedule.
- PaymentAmount: Number -- the scheduled payment amount. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 3.129 will result in a PaymentAmount of 3.13.
- StartDate: String -- the date on which the schedule will start running, not necessarily the date it will generate its first payment. Date format YYYY-MM-DD.
  NOTE: To make schedule management easier, it is strongly recommended that you set the schedule StartDate to the first day it will generate a payment.
- ExecutionFrequencyType: Integer-- the primary frequency on which to execute scheduled payments.
  enumeration: 1 = Daily, 2 = Weekly, 3 = BiWeekley, 4 = First of Month, 5 = Specific Day of Month,
  6 = Last of Month, 7 = Quarterly, 8 = Semi-Annually, 9 = Annually

Sample Request

<table>
<tr>
<td>Request:
{
"Request":{
"EndDate": "2018-07-01",
"PaymentAmount": 50,
"PaymentSubType": null,
"AccountId": 585373,
"InvoiceNumber": null,
"OrderId": null,
"FirstPaymentAmount": null,
"FirstPaymentDate": null,
"StartDate": "2017-07-01",
"ExecutionFrequencyType": 4,
"ExecutionFrequencyParameter": null,
"Description": "This is a recurring payment schedule entered via the API"
},
"Auth": {
"UserName": "APIUser***724",
"SecretAccessKey": "SecretAccessKey"</td>
<td>Response:
{
"Response": {
"Id": 161258,
"AccountId": 585373,
"CustomerId": "390839",
"PaymentAmount": 50,
"CustomerFirstName": "Patrick",
"CustomerLastName": "Exeter",
"CustomerCompany": null,
"PaymentSubType": 5,
"OrderId": null,
"InvoiceNumber": null,
"ExecutionFrequencyType": 4,
"ExecutionFrequencyParameter": 1,
"StartDate": "2017-07-01T06:00:00Z",
"EndDate": "2018-07-01T06:00:00Z",
"DateOfLastPaymentMade": null,
"NextScheduleDate": "2017-07-01T06:00:00Z",
"PauseUntilDate": null,
"TotalAmountPaid": 0,
"FirstPaymentAmount": 0,
"FirstPaymentDate": null,
"FirstPaymentDone": false,
"ScheduleStatus": 1,
"Description": "This is a recurring payment schedule entered via the API",
"CreatedOn": "2017-06-25T20:28:51.8928208Z",
"LastModified": "2017-06-25T20:28:51.8928208Z"
},
"Success": true,
"Message": null
}</td>
</tr>
</table>

Optional Attributes

- EndDate: String-- the date the schedule is to stop generating payments. If not included or left null, the schedule will default to running until manually disabled. Date format: YYYY-MM-DD.
- PaymentSubType: String-- this field defaults to “WEB” for ACH payments if not included or left null. enumeration (ACH): 0 = CCD; 2 = PPD; 4 = TEL; 5 = WEB
- InvoiceNumber: String-- 50 characters max; the Invoice Number attached to the schedule. If the number provided is a system Invoice Number, the associated system InvoiceId will be attached to the schedule and the scheduled payments will be applied to the system invoice.
  NOTE: To ensure that you do not collect more than the amount due for the system invoice it is better to use a Payment Plan schedule (instead of a recurring billing schedule) to discharge a system invoice.
  NOTE: Invoice creation and management is not available via the API.
- OrderId: String-- 50 characters max; a back-end transaction identifier not included in the UI.
- FirstPaymentAmount: Number-- a custom payment amount that is executed prior to the start of the schedule. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 23.129 will result in a FirstPaymentAmount of 23.13.
- FirstPaymentDate: String-- the date the custom first payment is to be processed. This can be the current date, or any future date, but must be prior to the schedule StartDate. If a future date is used, it can be edited prior to the custom first payment being generated. After the custom first payment is generated the field becomes read-only. Date format: YYYY-MM-DD.

- ExecutionFrequencyParameter: Integer -- used to specify the day of month for a Specific Day of Month schedule or to specify the day of the week for a weekly or bi-weekly schedule.
  enumeration (Weekly/Bi-Weekly): 1=Sunday; 2=Monday; 3=Tuesday; 4=Wednesday; 5=Thursday 6=Friday; 7=Saturday
  Monthly: Enter an integer from 1 -31
  NOTE: To avoid confusion, if you want to bill on the 30th or 31st, use a "Last Day of Month" Frequency Type instead of the "Specific Day of Month" Frequency Type.
- Description: String-- -- 2048 characters max.

Creating a new Recurring Schedule for an Existing Customer & new ACH Object

POST messages to endpoint +
/api/ReceivablesProPaymentSchedule/CreateNewAccountRecurringPayment

Required ACH Account Attributes

- CustomerId: Integer -- the system identifier for the Customer Record to which the card is being added. This is the CustomerId you retrieved from your database, or in the first step.
- RoutingNumber: String -- 9-digit bank routing number
- AccountNumber: String -- min 4 digits, max 100 digits; the bank account number
- BankName: String-- 100 characters max; the financial institution name
- IsCheckingAccount: Boolean -- enter "true" to indicate a checking account or "false" to indicate a savings account. General Ledger accounts are not supported.
- IsDefault: Boolean -- enter "true" to set the account as the default ACH Account for the Customer Record, otherwise enter "false." If there are no other ACH Accounts attached to the Customer Record, the field will default to "true" regardless of your entry.

Required Recurring Schedule Attributes

- PaymentAmount: Number -- the scheduled payment amount. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 3.129 will result in a PaymentAmount of 3.13.
- StartDate: String -- the date on which the schedule will start running, not necessarily the date it will generate its first payment. Date format YYYY-MM-DD.
  NOTE: To make schedule management easier, it is strongly recommended that you set the schedule StartDate to the first day it will generate a payment.
- ExecutionFrequencyType: Integer-- the primary frequency on which to execute scheduled payments.
  enumeration: 1 = Daily, 2 = Weekly, 3 = BiWeekley, 4 = First of Month, 5 = Specific Day of Month,
  6 = Last of Month, 7 = Quarterly, 8 = Semi-Annually, 9 = Annually

Sample Request

Request:

```
{
 "Request": {
  "Account": {
   "AccountNumber": "55214478",
   "RoutingNumber": "122100024",
   "BankName": "Test Bank",
   "IsCheckingAccount": true,
   "Id": 0,
   "CustomerId": 390927,
   "IsDefault": false,
   "CreatedOn": null,
   "LastModified": null
  },
  "RecurringPayment": {
  "EndDate": "2018-07-01",
  "PaymentAmount": 50,
  "PaymentSubType": null,
  "AccountId": 0,
  "InvoiceNumber": null,
  "OrderId": null,
  "FirstPaymentAmount": null,
  "FirstPaymentDate": null,
  "StartDate": "2017-07-01",
  "ExecutionFrequencyType": 4,
  "ExecutionFrequencyParameter": null,
  "Description": "This is a recurring payment schedule entered via
the API"
  }
 },
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
  "Response": {
    "Account": {
      "AccountNumber": "****4478",
      "RoutingNumber": "122100024",
      "BankName": "Test Bank",
      "IsCheckingAccount": true,
      "Id": 585421,
      "CustomerId": 390927,
      "IsDefault": false,
      "CreatedOn": "2017-06-25T21:10:23.6512565Z",
      "LastModified": "2017-06-25T21:10:23.6512565Z"
    },
    "RecurringPayment": {
      "Id": 161259,
      "AccountId": 585421,
      "CustomerId": "390927",
      "PaymentAmount": 50,
      "CustomerFirstName": "Axil",
      "CustomerLastName": "Mason",
      "CustomerCompany": "Axil's Axles",
      "PaymentSubType": 5,
      "OrderId": null,
      "InvoiceNumber": null,
      "ExecutionFrequencyType": 4,
      "ExecutionFrequencyParameter": 1,
      "StartDate": "2017-07-01T06:00:00Z",
      "EndDate": "2018-07-01T06:00:00Z",
      "DateOfLastPaymentMade": null,
      "NextScheduleDate": "2017-07-01T06:00:00Z",
      "PauseUntilDate": null,
      "TotalAmountPaid": 0,
      "FirstPaymentAmount": 0,
      "FirstPaymentDate": null,
      "FirstPaymentDone": false,
      "ScheduleStatus": 1,
      "Description": "This is a recurring payment schedule entered via
the API",
      "CreatedOn": "2017-06-25T21:10:23.4366948Z",
      "LastModified": "2017-06-25T21:10:23.4366948Z"
    }
  },
  "Success": true,
  "Message": null
}
```

Creating a new Recurring Schedule Object for a New Customer Object

POST messages to endpoint +
/api/ReceivablesProPaymentSchedule/CreateNewCustomerRecurringPayment

Required Attributes (Customer Object)

- FirstName: String -- 150 characters max
- LastName: String -- 150 characters max
- ShippingSameAsBilling: Boolean -- Set to “true” to indicate that there is no separate shipping address, even if you are not including a billing address. Set to “false” to require ShippingAddress fields.

Required Attributes (ACH Account Object)

- CustomerId: Integer -- the system identifier for the Customer Record to which the card is being added. This is the "Id" attribute from the Customer Object created in the previous step.
- RoutingNumber: String -- 9-digit bank routing number
- AccountNumber: String -- min 4 digits, max 100 digits; the bank account number
- BankName: String-- 100 characters max; the financial institution name
- IsCheckingAccount: Boolean -- enter "true" to indicate a checking account or "false" to indicate a savings account. General Ledger accounts are not supported.
- IsDefault: Boolean -- enter "true" to set the account as the default ACH Account for the Customer Record, otherwise enter "false." If there are no other ACH Accounts attached to the Customer Record, the field will default to "true" regardless of your entry.

Required Attributes (Recurring Schedule Object)

- AccountId: Integer -- the system identifier for the ACH Account used for the schedule. This is the "Id" attribute from the ACH Account Object created in the previously. The Customer Object associated with the AccountId will be associated with the schedule.
- PaymentAmount: Number -- the scheduled payment amount. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 3.129 will result in a PaymentAmount of 3.13.
- StartDate: String -- the date on which the schedule will start running, not necessarily the date it will generate its first payment. Date format YYYY-MM-DD.
  NOTE: To make schedule management easier, it is strongly recommended that you set the schedule StartDate to the first day it will generate a payment.
- ExecutionFrequencyType: Integer-- the primary frequency on which to execute scheduled payments.
  enumeration: 1 = Daily, 2 = Weekly, 3 = BiWeekley, 4 = First of Month, 5 = Specific Day of Month,
  6 = Last of Month, 7 = Quarterly, 8 = Semi-Annually, 9 = Annually

Sample Request

Request:


```
{
 "Request": {
   "Customer": {
    "Id": 0,
    "CustomerAccount": null,
    "FirstName": "Patricia",
    "LastName": "Giffort",
    "ShippingSameAsBilling": true
   },
   "Account": {
    "AccountNumber": "658874115",
    "RoutingNumber": "122100024",
    "BankName": "Test Bank",
    "IsCheckingAccount": true,
    "IsDefault": true,
   },
   "RecurringPayment": {
    "AccountId": 0,
    "CustomerId": 0,
    "PaymentAmount": 75,
    "PaymentSubType": null,
    "ExecutionFrequencyType": 5,
    "ExecutionFrequencyParameter": null,
    "StartDate": "2017-07-15",
    "EndDate": null,
   }
 },
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```


Response:


```
{
   "Response": {
      "Customer": {
         "Id": 391974,
         "CustomerAccount": null,
         "FirstName": "Patricia",
         "MiddleName": null,
         "LastName": "Giffort",
         "BillingAddress": null,
         "ShippingSameAsBilling": true,
         "ShippingAddress": null,
         "Company": null,
         "Phone": null,
         "AltPhone": null,
         "MobilePhone": null,
         "Fax": null,
         "Email": null,
         "AltEmail": null,
         "Website": null,
         "Notes": null,
         "CreatedOn": "2017-07-10T13:25:21.3452754Z",
         "LastModified": "2017-07-10T13:25:21.3452754Z"
      },
      "Account": {
         "AccountNumber": "*****4115",
         "RoutingNumber": "122100024",
         "BankName": "Test Bank",
         "IsCheckingAccount": true,
         "Id": 586546,
         "CustomerId": 391974,
         "IsDefault": true,
         "CreatedOn": "2017-07-10T13:25:21.8367327Z",
         "LastModified": "2017-07-10T13:25:21.8367327Z"
      },
      "RecurringPayment": {
         "Id": 161315,
         "AccountId": 586546,
         "CustomerId": "391974",
         "PaymentAmount": 75,
         "CustomerFirstName": "Patricia",
         "CustomerLastName": "Giffort",
         "CustomerCompany": null,
         "PaymentSubType": 5,
         "OrderId": null,
         "InvoiceNumber": null,
         "ExecutionFrequencyType": 4,
         "ExecutionFrequencyParameter": 1,
         "StartDate": "2017-07-15T06:00:00Z",
         "EndDate": null,
         "DateOfLastPaymentMade": null,
         "NextScheduleDate": "2017-08-01T06:00:00Z",
         "PauseUntilDate": null,
         "TotalAmountPaid": 0,
         "FirstPaymentAmount": 0,
         "FirstPaymentDate": null,
         "FirstPaymentDone": false,
         "ScheduleStatus": 1,
         "Description": null,
         "CreatedOn": "2017-07-10T13:25:21.8140099Z",
         "LastModified": "2017-07-10T13:25:21.8140099Z"
      }
   },
   "Success": true,
   "Message": null
}
```

Entering a One-time Payment for A Future Date

POST messages to endpoint + /api/ReceivablesProPaymentSchedule/CreateRecurringPayment

Entering a one-time payment to be executed on a future date is done by creating a recurring billing schedule and setting the start date to the payment date, the frequency to annual, and the end date to the day after the payment date.

Sample Request

Request:

```
{
 "Request":{
  "EndDate": "2018-07-07",
  "PaymentAmount": 25,
  "AccountId": 585421,
  "StartDate": "2017-07-06",
  "ExecutionFrequencyType": 9
},
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
}
```

Response:

```
{
  "Response": {
    "Id": 161260,
    "AccountId": 585421,
    "CustomerId": "390927",
    "PaymentAmount": 25,
    "CustomerFirstName": "Axil",
    "CustomerLastName": "Mason",
    "CustomerCompany": "Axil's Axles",
    "PaymentSubType": 5,
    "OrderId": null,
    "InvoiceNumber": null,
    "ExecutionFrequencyType": 9,
    "ExecutionFrequencyParameter": 0,
    "StartDate": "2017-07-06T06:00:00Z",
    "EndDate": "2018-07-07T06:00:00Z",
    "DateOfLastPaymentMade": null,
    "NextScheduleDate": "2017-07-06T06:00:00Z",
    "PauseUntilDate": null,
    "TotalAmountPaid": 0,
    "FirstPaymentAmount": 0,
    "FirstPaymentDate": null,
    "FirstPaymentDone": false,
    "ScheduleStatus": 1,
    "Description": null,
    "CreatedOn": "2017-06-25T22:23:26.1655731Z",
    "LastModified": "2017-06-25T22:23:26.1655731Z"
  },
  "Success": true,
  "Message": null
}
```

Editing an Existing Recurring Payment Schedule

You can use the API to make changes to Recurring Payment Schedules. The API does not support editing Payment Plan Schedules.

NOTE: For Payment Plans that have not yet started (StartDate is in the future) changes can be made via the UI. After a Payment Plan StartDate, changes are no longer permitted.

You can use the API to make changes to the following Recurring Payment Schedule fields:

- PaymentAmount: Number -- the scheduled payment amount. You can change this to any other number.
- AccountId: Integer -- the system identifier for the ACH Account used for the schedule. You can enter any other AccountId associated with the Customer Record here.
- PaymentSubType: String-- this field defaults to “WEB” if not included or left null. You can manually enter a new value for this field or omit it (or set it to null) and let the system assign the default sub-type for the AccountId.
- enumeration (ACH): 0 = CCD; 2 = PPD; 4 = TEL; 5 = WEB

- StartDate: String-- the date the schedule is configured to start running. If this date is in the future you can change it to the current date or any other future date. If this date has passed, then this field is read-only and cannot be changed. Date format: YYYY-MM-DD.
- EndDate: String-- the date the schedule is to stop generating payments. You can change this date to any other future date, or set it to "null" to run the schedule indefinitely. Date format: YYYY-MM-DD.
- ExecutionFrequencyType: Integer-- the primary frequency on which to execute scheduled payments. You can change this to any frequency.
- enumeration: 1 = Daily, 2 = Weekly, 3 = BiWeekley, 4 = First of Month, 5 = Specific Day of Month, 6 = Last of Month, 7 = Quarterly, 8 = Semi-Annually, 9 = Annually
  NOTE: If Quarterly, Semi-Annually, or Annually are used the schedule will execute the next payment on the Start Date (if the schedule has not yet started), or will calculate the next scheduled payment based on the date of the previous payment. For example, if you have a first of month schedule with a start date of January 1, 2016 that last generated a payment on September 1, 2016 and you change it to an Annual schedule, the next scheduled payment will be generated on September 1, 2017.
- ExecutionFrequencyParameter: Integer -- used to specify the day of month for a Specific Day of Month schedule or to specify the day of the week for a weekly or bi-weekly schedule. You can change this field whether or not you change the ExecutionFrequencyType, however the value used must be appropriate for the frequency type.
  enumeration (Weekly/Bi-Weekly): 1=Sunday; 2=Monday; 3=Tuesday; 4=Wednesday; 5=Thursday; 6=Friday; 7=Saturday
  Monthly: Enter an integer from 1 -31
  NOTE: To avoid confusion, if you want to bill on the 30th or 31st, use a "Last Day of Month" Frequency Type instead of the "Specific Day of Month" Frequency Type.
- FirstPaymentAmount: Number-- a custom payment amount that is executed prior to the start of the schedule. You can only change this amount if the FirstPaymentDate is in the future. Once the payment has been generated this field becomes read-only.
- FirstPaymentDate: String-- the date the custom first payment is to be processed. You can only change this if the existing date is in the future, and you can change it to the current date or any other future date. Once the payment has been generated this field becomes read-only. Date format: YYYY-MM-DD.
- InvoiceNumber: String-- 50 characters max; the Invoice Number attached to the schedule.
- OrderId: String-- 50 characters max; a back-end transaction identifier not included in the UI.
- Description: String-- -- 2048 characters max. An open text field.

NOTE: If changes are entered for any other fields, the system will ignore them and retain existing values.

Required Attributes

You must include the following Recurring Payment Object Required fields in your URI Request, whether or not you are changing their values:

- Id: Integer -- The ScheduleId for the Recurring Payment Object you are editing.
- PaymentAmount
- AccountId
- StartDate
- ExecutionFrequencyType (and ExecutionFrequencyParameter if required by the selected Type)

Sample Request

Request:

```
{
 "Request": {
  "Id": 161258,
  "AccountId": 585373,
  "PaymentAmount": 75,
  "ExecutionFrequencyType": 4,
  "StartDate": "2017-07-01"
 },

"Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
}
}
```

Response:

```
{
  "Response": {
    "Id": 161258,
    "AccountId": 585373,
    "CustomerId": "390839",
    "PaymentAmount": 75,
    "CustomerFirstName": "Patrick",
    "CustomerLastName": "Exeter",
    "CustomerCompany": null,
    "PaymentSubType": 5,
    "OrderId": null,
    "InvoiceNumber": null,
    "ExecutionFrequencyType": 4,
    "ExecutionFrequencyParameter": 0,
    "StartDate": "2017-07-01T06:00:00Z",
    "EndDate": null,
    "DateOfLastPaymentMade": null,
    "NextScheduleDate": "2017-07-01T06:00:00Z",
    "PauseUntilDate": null,
    "TotalAmountPaid": 0,
    "FirstPaymentAmount": 0,
    "FirstPaymentDate": null,
    "FirstPaymentDone": false,
    "ScheduleStatus": 1,
    "Description": null,
    "CreatedOn": "2017-06-25T20:28:51Z",
    "LastModified": "2017-06-25T22:57:14.1372032Z"
  },
  "Success": true,
  "Message": null
}
```

Suspending a Recurring Payment Schedule

POST messages to endpoint + /api/ReceivablesProPaymentSchedule/SuspendRecurringPayment

Suspending a schedule stops it from generating payments until it is manually resumed (see Payment Schedule for instructions). Any schedule (Recurring Payment or Payment Plan) with a Status of “Active” can be suspended via the API. Additionally, when the Suspend request is used with a paused schedule (see Payment Schedule), it will set the PauseUntilDate to null.

Sample Request

Request:

```
{
 "Request": 161072,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
  "Response": true,
  "Success": true,
  "Message": null
}
```

To see the updated status of the schedule, run a query to pull the latest information on the Payment Schedule to confirm the status has been changed to suspended (status = 4). This is not a required step, only a way to confirm the action.

Confirmation Request

(endpoint + /api/ReceivablesProPaymentSchedule/RecurringSchedule)

Request:


```
{
 "Request": 161072
,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```


Response:


```
{
  "Response": {
    "Id": 161072,
    "AccountId": 581025,
    "CustomerId": "385919",
    "PaymentAmount": 33.33,
    "CustomerFirstName": "Sabrina",
    "CustomerLastName": "Lewin",
    "CustomerCompany": null,
    "PaymentSubType": 2,
    "OrderId": null,
    "InvoiceNumber": null,
    "ExecutionFrequencyType": 4,
    "ExecutionFrequencyParameter": 1,
    "StartDate": "2017-06-01T06:00:00Z",
    "EndDate": null,
    "DateOfLastPaymentMade": "2017-06-01T06:00:00Z",
    "NextScheduleDate": "0001-01-01T07:00:00Z",
    "PauseUntilDate": null,
    "TotalAmountPaid": 33.33,
    "FirstPaymentAmount": 0,
    "FirstPaymentDate": null,
    "FirstPaymentDone": false,
    "ScheduleStatus": 4,
    "Description": null,
    "CreatedOn": "2017-05-03T17:26:50Z",
    "LastModified": "2017-06-25T23:07:53Z"
  },
  "Success": true,
  "Message": null
}
```


Pausing a Recurring Payment Schedule

POST messages to endpoint + /api/ReceivablesProPaymentSchedule/PauseRecurringPayment

Pausing a schedule stops it from generating payments until the date you specify. On the specified date, the schedule returns to “Active” status and will generate a payment on the next scheduled date, based on the schedule frequency.

For example, if a schedule with a First of Month frequency is paused until December 15, 2017 it will next process a payment on January 1, 2018.

Any schedule (Recurring Payment or Payment Plan) with a Status of “Active” can be paused via the API.

Sample Request

Request:


```
{
  "RecurringPaymentId": 161072,
 "EndDate": "2017-08-01",
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```


Response:


```
{
  "Response": true,
  "Success": true,
  "Message": null
}
```

To see the updated status of the schedule, run a query to pull the latest information on the Payment Schedule to confirm the status has been changed to suspended (status = 4). This is not a required step, only a way to confirm the action.

Confirmation Request

(endpoint + /api/ReceivablesProPaymentSchedule/RecurringSchedule)

Request:

```
{
  "RecurringPaymentId": 161072,
 "EndDate": "2017-08-01",
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
   "Response": {
      "Id": 161072,
      "AccountId": 581025,
      "CustomerId": "385919",
      "PaymentAmount": 33.33,
      "CustomerFirstName": "Sabrina",
      "CustomerLastName": "Lewin",
      "CustomerCompany": null,
      "PaymentSubType": 2,
      "OrderId": null,
      "InvoiceNumber": null,
      "ExecutionFrequencyType": 4,
      "ExecutionFrequencyParameter": 1,
      "StartDate": "2017-06-01T06:00:00Z",
      "EndDate": null,
      "DateOfLastPaymentMade": "2017-06-01T06:00:00Z",
      "NextScheduleDate": "2017-08-01T06:00:00Z",
      "PauseUntilDate": "2017-08-01T06:00:00Z",
      "TotalAmountPaid": 33.33,
      "FirstPaymentAmount": 0,
      "FirstPaymentDate": null,
      "FirstPaymentDone": false,
      "ScheduleStatus": 4,
      "Description": null,
      "CreatedOn": "2017-05-03T17:26:50Z",
      "LastModified": "2017-06-25T23:21:42Z"
   },
   "Success": true,
   "Message": null
}
```

Resuming a Recurring Payment Schedule

POST messages to endpoint + /api/ReceivablesProPaymentSchedule/ResumeRecurringPayment

You can resume any “Suspended” schedule, whether suspended indefinitely or paused until a specific date, using the resume schedule feature.

Sample Request

Request:

```
{
 "Request": 161072
,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
   "Response": true,
   "Success": true,
   "Message": null
}
```

To see the updated status of the schedule, run a query to pull the latest information on the Payment Schedule to confirm the status has been changed to active (status = 1). This is not a required step, only a way to confirm the action.

Confirmation Request

(endpoint + /api/ReceivablesProPaymentSchedule/RecurringSchedule)

Request:


```
{
 "Request": 161072
,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
} }
}
```


Response:


```
{
  "Response": {
    "Id": 161072,
    "AccountId": 581025,
    "CustomerId": "385919",
    "PaymentAmount": 33.33,
    "CustomerFirstName": "Sabrina",
    "CustomerLastName": "Lewin",
    "CustomerCompany": null,
    "PaymentSubType": 2,
    "OrderId": null,
    "InvoiceNumber": null,
    "ExecutionFrequencyType": 4,
    "ExecutionFrequencyParameter": 1,
    "StartDate": "2017-06-01T06:00:00Z",
    "EndDate": null,
    "DateOfLastPaymentMade": "2017-06-01T06:00:00Z",
    "NextScheduleDate": "2017-07-01T06:00:00Z",
    "PauseUntilDate": null,
    "TotalAmountPaid": 33.33,
    "FirstPaymentAmount": 0,
    "FirstPaymentDate": null,
    "FirstPaymentDone": false,
    "ScheduleStatus": 1,
    "Description": null,
    "CreatedOn": "2017-05-03T17:26:50Z",
    "LastModified": "2017-06-25T23:14:47Z"
  },
  "Success": true,
  "Message": null
}
```


Deleting a Payment Schedule

POST messages to endpoint + /api/ReceivablesProPaymentSchedule/RemoveRecurringPayment

You can delete any payment schedule (Recurring Payment or Payment Plan) that has not yet started and does not have a transaction attached to it. This means that neither the custom first payment nor any regularly scheduled payments have yet been generated by the schedule, and that the schedule StartDate is not in the past. (Schedules with a StartDate on the current date can be deleted.)

Sample Request

Request:


```
{
 "Request": 161260,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```


Response:


```
{
  "Response": true,
  "Success": true,
  "Message": null
}
```

If you request to delete a schedule that cannot be deleted (either because it has a StartDate in the past or has a transaction attached to it), you will see a response like the one below:

Request:	Response:

{
"Request": 161080,
"Auth": {
"UserName": "APIUser***724",
"SecretAccessKey": "SecretAccessKey"
}
}
	{
	"Response": false,
	"Success": false,
	"Message": "Endpoint Error: BadRequest: StatusCode: 400 ErrorCode: InvalidInput TraceCode: ErrorMessages: - Payments have been made on this schedule. Payment schedules that have started cannot be deleted, only suspended"
	}

If there is a need to halt payments on a schedule that has already begun, the schedule should be indefinitely suspended rather than deleted.

Payment Plan Objects

Payment Plan Object Response Attributes

- PaymentScheduleType: String-- "RecurringPayment" for a Recurring Payment schedule or "PaymentPlan for a Payment Plan schedule.
- CustomerFirstName: String -- the first name of the Customer associated with the payment.
- CustomerLastName: String -- the last name of the Customer associated with the payment.
- CustomerId: Integer -- the system identifier for the Customer Record associated with the payment.
- ScheduleStatus: Integer -- the current schedule Status.
  Enumeration: 1 = Active; 3 = Expired; 4 = Suspended
- NextPaymentDate: String -- the next date on which the schedule will generate a payment, based on the schedule frequency. Date format YYYY-MM-DDT00:00:00Z. (You can ignore the time portion of this date.)
- PaymentAmount: Number-- the amount of each scheduled payment.
- Id: Integer -- this is the ScheduleId number you will need to reference in future calls related to this schedule including pause, suspend, resume, and view details.
- LastModified: String -- set by the system, the date and time the schedule was last modified.
- CreatedOn: String -- set by the system, the date and time the schedule object was created.

Creating a new Payment Plan Object for an Existing Customer & ACH Object

POST messages to endpoint + /api/ReceivablesProPaymentSchedule/CreateRecurringPayment

Required Attributes

- AccountId: Integer -- the system identifier for the ACH Account used for the schedule. This is the "Id" attribute from the ACH Account Object created in the previous step. The Customer Object associated with the AccountId will be associated with the schedule.
- TotalDueAmount: Number -- the total amount to be collected for the schedule. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 300.129 will result in a TotalDueAmount of 300.13.
- TotalNumberOfPayments: Integer -- the total number of schedule payments. Enter any integer between 1 and 99.
- StartDate: String -- the date on which the schedule will start running, not necessarily the date it will generate its first payment. Date format YYYY-MM-DD. Date format: YYYY-MM-DD.
- NOTE: To make schedule management easier, it is strongly recommended that you set the schedule StartDate to the first day it will generate a payment.
- ExecutionFrequencyType: Integer-- the primary frequency on which to execute scheduled payments. enumeration: 1 = Daily, 2 = Weekly, 3 = BiWeekley, 4 = First of Month, 5 = Specific Day of Month, 6 = Last of Month, 7 = Quarterly, 8 = Semi-Annually, 9 = Annually

Sample Request

Request:


```
{
  "Request":{
    "EndDate": null,
    "PaymentAmount": 200,
    "PaymentSubType": 2,
    "AccountId": 585373,
    "InvoiceNumber": null,
    "OrderId": null,
    "FirstPaymentAmount": null,
    "FirstPaymentDate": null,
    "StartDate": "2017-07-15",
    "ExecutionFrequencyType": 4,
    "ExecutionFrequencyParameter": null,
    "Description": "This is a recurring payment schedule entered via the API"
  },

  "Auth": {
    "UserName": "APIUser***724",
    "SecretAccessKey": "SecretAccessKey"
  }
```


Response:


```
{
    "Response": {
        "Id": 161319,
        "AccountId": 585373,
        "CustomerId": "390839",
        "PaymentAmount": 200,
        "CustomerFirstName": "Patrick",
        "CustomerLastName": "Exeter",
        "CustomerCompany": null,
        "PaymentSubType": 2,
        "OrderId": null,
        "InvoiceNumber": null,
        "ExecutionFrequencyType": 4,
        "ExecutionFrequencyParameter": 1,
        "StartDate": "2017-07-15T06:00:00Z",
        "EndDate": null,
        "DateOfLastPaymentMade": null,
        "NextScheduleDate": "2017-08-01T06:00:00Z",
        "PauseUntilDate": null,
        "TotalAmountPaid": 0,
        "FirstPaymentAmount": 0,
        "FirstPaymentDate": null,
        "FirstPaymentDone": false,
        "ScheduleStatus": 1,
        "Description": "This is a recurring payment schedule entered via the API",
        "CreatedOn": "2017-07-10T13:37:40.3146491Z",
        "LastModified": "2017-07-10T13:37:40.3146491Z"
    },
    "Success": true,
    "Message": null
}
```

Optional Attributes

- PaymentSubType: String-- this field defaults to "WEB" for ACH payments if not included or left null. enumeration (ACH): 0 = CCD; 2 = PPD; 4 = TEL; 5 = WEB
- InvoiceNumber: String-- 50 characters max; the Invoice Number attached to the schedule. If the number provided is a system Invoice Number, the associated system InvoiceId will be attached to the schedule and the scheduled payments will be applied to the system invoice.
  NOTE: Invoice creation and management is not available via the API.
- OrderId: String-- 50 characters max; a back-end transaction identifier not included in the UI.
- FirstPaymentAmount: Number-- a custom payment amount that is executed prior to the start of the schedule. Enter a number only, do not include the $. You can enter an integer or a decimal. If you use more than 2 decimal places, the system will round to the nearest penny-- i.e. entering 23.129 will result in a FirstPaymentAmount of 23.13.
- FirstPaymentDate: String-- the date the custom first payment is to be processed. This can be the current date, or any future date, but must be prior to the schedule StartDate. If a future date is used, it can be edited prior to the custom first payment being generated. After the custom first payment is generated the field becomes read-only. Date format: YYYY-MM-DD.
- ExecutionFrequencyParameter: Integer -- used to specify the day of month for a Specific Day of Month schedule or to specify the day of the week for a weekly or bi-weekly schedule.
  enumeration (Weekly/Bi-Weekly): 1=Sunday; 2=Monday; 3=Tuesday; 4=Wednesday; 5=Thursday 6=Friday; 7=Saturday
  Monthly: Enter an integer from 1 -31
  NOTE: To avoid confusion, if you want to bill on the 30th or 31st, use a "Last Day of Month" Frequency Type instead of the "Specific Day of Month" Frequency Type.
- Description: String-- -- 2048 characters max.

Retrieving Payment Details

Single Transaction Query

To retrieve details about a single payment, use the Transaction ID generated by the ReceivablesPro system in the request. Single payment object requests will always display in "complete" form, the "lite" version is not available.

POST messages to endpoint + /api/ReceivablesProPayment/Payment

Sample Request

Request:

```
{
 "Request": 4511411,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
  "Response": {
    "Id": 4511411,
    "AccountId": 581242,
    "Amount": 100,
    "IsDebit": true,
    "Cvv": null,
    "PaymentSubType": 2,
    "InvoiceId": null,
    "InvoiceNumber": "",
    "PurchaseOrderNumber": "",
    "OrderId": null,
    "Description": "",
    "Latitude": null,
    "Longitude": null,
    "SuccessReceiptOptions": null,
    "FailureReceiptOptions": null,
    "CustomerId": 386124,
    "CustomerFirstName": "Rex",
    "CustomerLastName": "Winchester",
```

	"CustomerCompany": null, "ReferenceId": 4505629, "Status": 9, "RecurringScheduleId": 0, "PaymentType": 2, "ProviderAuthCode": "Approved", "TraceNumber": "64379ade-f4c2-4f4a-a31e-e280d4e75328", "PaymentDate": "2017-06-25T06:00:00Z", "ReturnDate": null, "EstimatedSettleDate": "2017-06-27T06:00:00Z", "ActualSettleDate": null, "CanVoidUntil": "2017-06-25T21:00:00Z", "FailureData": null, "IsDecline": false, "CreatedOn": "2017-06-25T18:30:31Z", "LastModified": "2017-06-25T18:30:31Z", "RequiresReceipt": false, "PaymentToken": null }, "Success": true, "Message": null }

Multiple Transaction Query

The API can be used to return a list of all payments processed. The generated list can be in complete format or "lite" format, which is ideal for creating summary tables and snapshots.

POST messages to endpoint + /api/ReceivablesProPayment/Payments

Because the full list of transactions can grow quite large and unwieldy, the API uses paging to limit the number of results returned from a single Request. The default is to show the most recent 200 transactions. You can use Pagination filters to govern both the number of results returned in the Response as well as the Response page you would like to retrieve.

Pagination filters

Page = the set of results you want to return. "1" returns the first page, "2" the second, etc. If this filter is not included, the Request defaults to "1."

Pagesize = the number of payment records per page. If this filter is not included, the Request defaults to "200."

Status Filter

When returning a list of transactions, you can filter that list by status. You can generate lists that include transactions of a single status, or a group of one or more statuses. The default is to return transactions of all statuses.

Code	Description
0	Authorized
1	Chargeback
2*	Failed
3	Pending
4*	Posted
5	RefundSettled
6	Returned
7	Reversed
8	ReverseNSF

Date Filters

When requesting a transaction list, you can use date filters to return transactions for a given date range. The default is to return all transactions, regardless of date.

The Date Format is: YYYY-MM-DD

startdate = set this filter to return all transactions processed on or after the given date.

enddate= set this filter to return all transactions processed on or before the given date.

List Sorting

When returning a list of transactions in complete or "lite" form you can use "SortBy" to control the Attribute by which the list is sorted, and whether the sort is in ascending or descending order. The default sort is Descending by PaymentId. This results in listing the most recent transactions first.

Code	Description
0	ActualSettledDate
1	Amount
2	EstimatedSettleDate
3	PaymentDate
4	PaymentId
5	PaymentSubType
6	PaymentType
7	ReturnDate

"Direction" controls if the list is sorted in ascending order ("0") or descending order ("1").

Sample Request

Request:

```
{
 "StartDate": "2017-06-20",
 "EndDate": "2017-06-26",
 "Status": [
  4
 ],
 "SortBy": 1,
 "Direction": 1,
 "Page": 1,
 "PageSize": 3,
 "Lite": true

"Auth": {
   "UserName": "APIUser***724",
   "SecretAccessKey": "SecretAccessKey"
 }
}
```

Response:

```
{
  "Response": {
     "Page": 1,
     "ItemsPerPage": 3,
     "TotalItems": 12,
     "TotalPages": 4,
     "Items": [
        {
           "Id": 4511401,
           "AccountId": null,
           "Amount": 200,
           "IsDebit": false,
           "Cvv": null,
           "PaymentSubType": null,
           "InvoiceId": null,
           "InvoiceNumber": null,
           "PurchaseOrderNumber": null,
           "OrderId": null,
           "Description": null,
           "Latitude": null,
           "Longitude": null,
           "SuccessReceiptOptions": null,
           "FailureReceiptOptions": null,
           "CustomerId": 0,
           "CustomerFirstName": "Rex",
           "CustomerLastName": "Winchester",
           "CustomerCompany": null,
           "ReferenceId": 0,
```

```
    "Status": 4,
    "RecurringScheduleId": 0,
    "PaymentType": 0,
    "ProviderAuthCode": null,
    "TraceNumber": null,
    "PaymentDate": "2017-06-25T06:00:00Z",
    "ReturnDate": null,
    "EstimatedSettleDate": null,
    "ActualSettleDate": null,
    "CanVoidUntil": null,
    "FailureData": null,
    "IsDecline": false,
    "CreatedOn": null,
    "LastModified": null,
    "RequiresReceipt": false,
    "PaymentToken": null
  },
  {
    "Id": 4511400,
    "AccountId": null,
    "Amount": 100,
    "IsDebit": false,
    "Cvv": null,
    "PaymentSubType": null,
    "InvoiceId": null,
    "InvoiceNumber": null,
    "PurchaseOrderNumber": null,
    "OrderId": null,
    "Description": null,
    "Latitude": null,
    "Longitude": null,
    "SuccessReceiptOptions": null,
    "FailureReceiptOptions": null,
    "CustomerId": 0,
    "CustomerFirstName": "Rex",
    "CustomerLastName": "Winchester",
    "CustomerCompany": null,
    "ReferenceId": 0,
    "Status": 4,
    "RecurringScheduleId": 0,
    "PaymentType": 0,
    "ProviderAuthCode": null,
    "TraceNumber": null,
    "PaymentDate": "2017-06-25T06:00:00Z",
    "ReturnDate": null,
    "EstimatedSettleDate": null,
    "ActualSettleDate": null,
    "CanVoidUntil": null,
    "FailureData": null,
    "IsDecline": false,
    "CreatedOn": null,
    "LastModified": null,
    "RequiresReceipt": false,
    "PaymentToken": null
  },
  {
    "Id": 4511399,
    "AccountId": null,
    "Amount": 50,
    "IsDebit": false,
    "Cvv": null,
    "PaymentSubType": null,
    "InvoiceId": null,
    "InvoiceNumber": null,
    "PurchaseOrderNumber": null,
    "OrderId": null,
```

```
            "Description": null,
            "Latitude": null,
            "Longitude": null,
            "SuccessReceiptOptions": null,
            "FailureReceiptOptions": null,
            "CustomerId": 0,
            "CustomerFirstName": "Rex",
            "CustomerLastName": "Winchester",
            "CustomerCompany": null,
            "ReferenceId": 0,
            "Status": 4,
            "RecurringScheduleId": 0,
            "PaymentType": 0,
            "ProviderAuthCode": null,
            "TraceNumber": null,
            "PaymentDate": "2017-06-25T06:00:00Z",
            "ReturnDate": null,
            "EstimatedSettleDate": null,
            "ActualSettleDate": null,
            "CanVoidUntil": null,
            "FailureData": null,
            "IsDecline": false,
            "CreatedOn": null,
            "LastModified": null,
            "RequiresReceipt": false,
            "PaymentToken": null
        }
    ]
  },
  "Success": true,
  "Message": null
}
```


Retrieving Payments for a Recurring Payment Schedule Object

POST message to endpoint + /api/ReceivablesProPaymentSchedule/RecurringPayments

This will return settled payments for a specific Schedule ID.

Sample Request

Request:


```
{
 "Request": 161154,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecredAccessKey"
 }
}
```


Response:


```
{
  "Response": {
    "Page": 1,
    "ItemsPerPage": 200,
    "TotalItems": 3,
    "TotalPages": 1,
    "Items": [
      {
        "Id": 4475105,
        "AccountId": 582082,
        "Amount": 32,
        "IsDebit": false,
        "Cvv": null,
        "PaymentSubType": 4,
        "InvoiceId": null,
        "InvoiceNumber": "",
        "PurchaseOrderNumber": "",
        "OrderId": null,
        "Description": "",
        "Latitude": null,
        "Longitude": null,
        "SuccessReceiptOptions": null,
        "FailureReceiptOptions": null,
```

```
            "CustomerId": 387505,
            "CustomerFirstName": "Jeannie ",
            "CustomerLastName": "Loman",
            "CustomerCompany": "",
            "ReferenceId": 0,
            "Status": 10,
            "RecurringScheduleId": 161154,
            "PaymentType": 2,
            "ProviderAuthCode": "",
            "TraceNumber": "984022b6-1302-43b8-81bb-
0b058fbd62b8",
            "PaymentDate": "2017-05-19T06:00:00Z",
            "ReturnDate": null,
            "EstimatedSettleDate": "2017-05-24T06:00:00Z",
            "ActualSettleDate": null,
            "CanVoidUntil": "2017-05-19T21:00:00Z",
            "FailureData": {
               "Code": null,
               "Description": null,
               "MerchantActionText": null,
               "IsDecline": false
            },
            "IsDecline": false,
            "CreatedOn": "2017-05-19T07:30:59Z",
            "LastModified": "2017-05-20T10:00:07Z",
            "RequiresReceipt": false,
            "PaymentToken": null
         },
         {
            "Id": 4475782,
            "AccountId": 582082,
            "Amount": 25,
            "IsDebit": false,
            "Cvv": null,
            "PaymentSubType": 4,
            "InvoiceId": null,
            "InvoiceNumber": "",
            "PurchaseOrderNumber": "",
            "OrderId": null,
            "Description": "",
            "Latitude": null,
            "Longitude": null,
            "SuccessReceiptOptions": null,
            "FailureReceiptOptions": null,
            "CustomerId": 387505,
            "CustomerFirstName": "Jeannie ",
            "CustomerLastName": "Loman",
            "CustomerCompany": "",
            "ReferenceId": 0,
            "Status": 10,
            "RecurringScheduleId": 161154,
            "PaymentType": 2,
            "ProviderAuthCode": "",
            "TraceNumber": "c7ab23ff-9b3c-4478-a959-63e6f460b007",
            "PaymentDate": "2017-05-20T06:00:00Z",
            "ReturnDate": null,
            "EstimatedSettleDate": "2017-05-25T06:00:00Z",
            "ActualSettleDate": null,
            "CanVoidUntil": "2017-05-20T21:00:00Z",
            "FailureData": {
               "Code": null,
               "Description": null,
               "MerchantActionText": null,
               "IsDecline": false
            },
            "IsDecline": false,
```

```
            "CreatedOn": "2017-05-20T07:31:02Z",
            "LastModified": "2017-05-21T10:00:07Z",
            "RequiresReceipt": false,
            "PaymentToken": null
         },
         {
            "Id": 4507709,
            "AccountId": 582082,
            "Amount": 25,
            "IsDebit": false,
            "Cvv": null,
            "PaymentSubType": 4,
            "InvoiceId": null,
            "InvoiceNumber": "",
            "PurchaseOrderNumber": "",
            "OrderId": null,
            "Description": "",
            "Latitude": null,
            "Longitude": null,
            "SuccessReceiptOptions": null,
            "FailureReceiptOptions": null,
            "CustomerId": 387505,
            "CustomerFirstName": "Jeannie ",
            "CustomerLastName": "Loman",
            "CustomerCompany": "",
            "ReferenceId": 0,
            "Status": 10,
            "RecurringScheduleId": 161154,
            "PaymentType": 2,
            "ProviderAuthCode": "",
            "TraceNumber": "75798658-9fb5-4d4e-be24-
285829648274",
            "PaymentDate": "2017-06-20T06:00:00Z",
            "ReturnDate": null,
            "EstimatedSettleDate": "2017-06-23T06:00:00Z",
            "ActualSettleDate": null,
            "CanVoidUntil": "2017-06-20T21:00:00Z",
            "FailureData": {
               "Code": null,
               "Description": null,
               "MerchantActionText": null,
               "IsDecline": false
            },
            "IsDecline": false,
            "CreatedOn": "2017-06-20T07:31:00Z",
            "LastModified": "2017-06-21T10:00:06Z",
            "RequiresReceipt": false,
            "PaymentToken": null
         }
      ]
   },
   "Success": true,
   "Message": null
}
```

Retrieving Payments for a Payment Plan Schedule Object

POST messages to endpoint + /api/ReceivablesProPaymentSchedule/PaymentPlanPayments

This will return settled payments for a specific Schedule ID.

Sample Request

Request:

```
{
 "Request": 161244,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecredAccessKey"
 }
}
```

Response:


```
{
  "Response": {
    "Page": 1,
    "ItemsPerPage": 200,
    "TotalItems": 1,
    "TotalPages": 1,
    "Items": [
      {
        "Id": 4507290,
        "AccountId": 584641,
        "Amount": 25,
        "IsDebit": false,
        "Cvv": null,
        "PaymentSubType": 2,
        "InvoiceId": null,
        "InvoiceNumber": "",
        "PurchaseOrderNumber": "",
        "OrderId": null,
        "Description": "",
        "Latitude": null,
        "Longitude": null,
        "SuccessReceiptOptions": null,
        "FailureReceiptOptions": null,
        "CustomerId": 390217,
        "CustomerFirstName": "Robert",
        "CustomerLastName": "Vance",
        "CustomerCompany": "Vance Refrigeration",
        "ReferenceId": 0,
        "Status": 10,
        "RecurringScheduleId": 161244,
        "PaymentType": 2,
        "ProviderAuthCode": "",
        "TraceNumber": "3f4c2a5a-24c4-4c50-a213-e4a04f4b4ee9",
        "PaymentDate": "2017-06-19T06:00:00Z",
        "ReturnDate": null,
        "EstimatedSettleDate": "2017-06-22T06:00:00Z",
        "ActualSettleDate": null,
        "CanVoidUntil": "2017-06-19T21:00:00Z",
        "FailureData": {
          "Code": null,
          "Description": null,
          "MerchantActionText": null,
          "IsDecline": false
        },
        "IsDecline": false,
        "CreatedOn": "2017-06-19T18:00:07Z",
        "LastModified": "2017-06-20T10:00:08Z",
        "RequiresReceipt": false,
        "PaymentToken": null
      }
    ]
  },
  "Success": true,
  "Message": null
}
```

Retrieving Information on All Schedules (Recurring & Payment Plans)

POST messages to endpoint + /api/ReceivablesProPaymentSchedule/AllPaymentSchedules

Returns abbreviated records for all schedules (including expired and suspended schedules). The default is to list schedules from oldest to newest.

If you want the detail record for one of the schedules, note its type and ID, and then retrieve the full schedule. (See Payment Schedule for instructions.)

Sample Request (results truncated)

Request:


```
{
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecredAccessKey"
 }
}
```


Response:


```
{
  "Response": {
    "Page": 1,
    "ItemsPerPage": 200,
    "TotalItems": 33,
    "TotalPages": 1,
    "Items": [
      {
        "Id": 160795,
        "AccountId": 0,
        "CustomerId": "380118",
        "PaymentAmount": 18.75,
        "CustomerFirstName": "Stephen",
        "CustomerLastName": "Tennison",
        "CustomerCompany": null,
        "PaymentSubType": null,
        "OrderId": null,
        "InvoiceNumber": null,
        "ExecutionFrequencyType": 0,
        "ExecutionFrequencyParameter": 0,
        "StartDate": null,
        "EndDate": null,
        "DateOfLastPaymentMade": null,
        "NextScheduleDate": null,
        "PauseUntilDate": null,
        "TotalAmountPaid": 0,
        "FirstPaymentAmount": null,
        "FirstPaymentDate": null,
        "FirstPaymentDone": false,
        "ScheduleStatus": 3,
        "Description": null,
        "CreatedOn": "2017-03-06T22:48:03Z",
        "LastModified": "2017-03-28T07:30:51Z"
      },
      {
        "Id": 160797,
        "AccountId": 0,
        "CustomerId": "380080",
        "PaymentAmount": 150,
        "CustomerFirstName": "Keefe",
        "CustomerLastName": "Stamp",
        "CustomerCompany": null,
        "PaymentSubType": null,
        "OrderId": null,
        "InvoiceNumber": null,
        "ExecutionFrequencyType": 0,
        "ExecutionFrequencyParameter": 0,
        "StartDate": null,
        "EndDate": null,
        "DateOfLastPaymentMade": null,
        "NextScheduleDate": null,
        "PauseUntilDate": null,
```

```
            "TotalAmountPaid": 0,
            "FirstPaymentAmount": null,
            "FirstPaymentDate": null,
            "FirstPaymentDone": false,
            "ScheduleStatus": 3,
            "Description": null,
            "CreatedOn": "2017-03-07T21:59:55Z",
            "LastModified": "2017-03-09T08:30:58Z"
        },
        {
            "Id": 160799,
            "AccountId": 0,
            "CustomerId": "380161",
            "PaymentAmount": 12.5,
            "CustomerFirstName": "Beverly",
            "CustomerLastName": "Alger",
            "CustomerCompany": null,
            "PaymentSubType": null,
            "OrderId": null,
            "InvoiceNumber": null,
            "ExecutionFrequencyType": 0,
            "ExecutionFrequencyParameter": 0,
            "StartDate": null,
            "EndDate": null,
            "DateOfLastPaymentMade": null,
            "NextScheduleDate": null,
            "PauseUntilDate": null,
            "TotalAmountPaid": 0,
            "FirstPaymentAmount": null,
            "FirstPaymentDate": null,
            "FirstPaymentDone": false,
            "ScheduleStatus": 3,
            "Description": null,
            "CreatedOn": "2017-03-08T23:26:54Z",
            "LastModified": "2017-03-22T07:31:00Z"
                }
    ]
},
"Success": true,
"Message": null
```

Retrieving Payment Schedules Associated with A Customer

POST messages to endpoint + /api/ReceivablesProCustomer/PaymentSchedules

Use to return the set of schedules attached to a single customer record. This list can be returned in either full or "lite" format. For detailed descriptions of schedule object attributes see Payment Schedule Response Object Attributes, Recurring Payment Response Object Attributes and Payment Plan Response.

Sample Request

Request:


```
{
 "CustomerId": 387505,
 "StartDate": "2017-02-01",
 "EndDate": "2017-06-26",
 "Status": 1,
 "SortBy": 1,
 "Direction": 0,
 "Page": 1,
 "PageSize": 100,
 "Lite": false,
 "Auth": {
  "UserName": "APIUser***724",
  "SecretAccessKey": "SecretAccessKey"
 }
}
```


Response:


```
{
  "Response": {
    "Page": 1,
    "ItemsPerPage": 100,
    "TotalItems": 1,
    "TotalPages": 1,
    "Items": {
      "PaymentPlans": [],
      "RecurringPayments": [
        {
          "Id": 161154,
          "AccountId": 582082,
          "CustomerId": "387505",
          "PaymentAmount": 25,
          "CustomerFirstName": "Jeannie ",
          "CustomerLastName": "Loman",
          "CustomerCompany": "",
          "PaymentSubType": 4,
          "OrderId": null,
          "InvoiceNumber": "",
          "ExecutionFrequencyType": 5,
          "ExecutionFrequencyParameter": 20,
          "StartDate": "2017-05-20T06:00:00Z",
          "EndDate": "2017-08-20T06:00:00Z",
          "DateOfLastPaymentMade": "2017-06-20T06:00:00Z",
          "NextScheduleDate": "2017-07-20T06:00:00Z",
          "PauseUntilDate": null,
          "TotalAmountPaid": 82,
          "FirstPaymentAmount": 32,
          "FirstPaymentDate": "2017-05-19T06:00:00Z",
          "FirstPaymentDone": true,
          "ScheduleStatus": 1,
          "Description": "",
          "CreatedOn": "2017-05-15T20:56:26Z",
          "LastModified": "2017-06-20T07:31:00Z"
        }
      ]
    }
  },
  "Success": true,
  "Message": null
}
```

Payliance Failure Codes

CODE	DESCRIPTION	MERCHANT ACTION TEXT
4502		
4504	Merchant's maximum transaction amount exceeded	Contact customer service.
4510	Invalid Merchant ID or Password	Contact customer service.
4513	Invalid Routing Number	Re-enter the payment information or try a different payment method.
4516	Unauthorized SEC type	Contact customer service.