PaySimple®

# API Developer Reference

## Status and Version

Last updated: Wednesday, May 18, 2011

## This Document

This document describes the PaySimple Application Programming Interface (API) and associated services.

## Intended Audience

The intended audience for this document is programmers with a basic understanding of application programming interfaces and standards such as Simple Object Access Protocol (SOAP), XML, JavaScript Object Notation (JSON) and Web Services Definition Language (WSDL).

## Revision History

| DATE | DESCRIPTION |
|---|---|
| May 20, 2010 | Document creation |
| August 11, 2010 | Added documentation on optional payment parameters<br>Added PsScheduleStatus, PsInvoiceScheduleStatus, and PsInvoiceStatus enumerations<br>Added GetinvoiceById, ListRecurringPayments, ListScheduledInvoices methods |
| September 20, 2010 | Removing reference to 3.0rc and updating links to use 3.00 |
| September 29,2010 | Updating verbiage |
| October 4, 2010 | Updating help links |
| November 26, 2010 | Added ListPaymentsByRecurringSchedule method<br>Added ListPaymentsByInvoiceId method<br>Added SuspendRecurringPaymentSchedule method<br>Added ResumeRecurringPaymentSchedule method<br>Added ListPaymentsEx method<br>Added ListCustomerWithSort method |

| | |
|---|---|
| | Added PsCustomerPagedCriteria object<br>Added PsPaymentPagedCriteria object<br>Added PsPaymentStatusFilter object<br>Added PaymentStatus enumeration<br>Added PsPaymentDateRangeType enumeration |
| December 28, 2010 | Updated PsPaymentPagedCriteria definition<br>Added Sort enumeration<br>Added GetRecurringPaymentSchedule method<br>Added GetDefaultCustomerAccount method<br>Added SetDefaultCustomerAccount method<br>Added PsDefaultCustomerAccount object |
| February 1, 2011 | Added SearchCustomerEx method<br>Added ModifyRecurringPaymentSchedule method<br>Added DeleteRecurringSchedule method<br>Added GetDefaultCreditCardAccount method<br>Added  GetDefaultAchAccount method<br>Added GetCustomerByConsumerData |
| May 18, 2011 | Added PsUser, PsInvoiceEx, and PsInvoiceLineItemEx<br>Added AddInvoiceEx, GetInvoiceByIdEx, GetUserInfo, ResendInvoice, GetCustomerAndDefaultAccounts, GetCustomerAccountByAccountId, ListAllCustomerAccounts methods |

# Table of Contents

# Introduction

The PaySimple API allows merchants programmatic access to PaySimple services such as payment processing, invoicing and reporting.  PaySimple bases its API on known open standards such as Simple Object Access Protocol (SOAP), JavaScript Object Notation (JSON), Web Services Definition Language (WSDL), and XML.

# Implementation Overview

The PaySimple API is versioned allowing PaySimple to upgrade the API without breaking previous implementations. The strategy for this approach is seen in the service endpoint url's. Sandbox endpoints are of the form https://sandbox-api.paysimple.com/{version}/paysimpleapi/json while the production endpoints are of the form https://api.paysimple.com/{version}/paysimpleapi/json.

There are 3 types of implementation options to choose from: SOAP, XML, and JSON. The endpoints for each service are below.

PaySimple provides a sandbox environment for development and testing.  Once approved, authentication keys will be provided to migrate into the live production environment.

## SOAP

Sandbox environment: The sandbox WSDL for SOAP implementations resides at https://sandbox-api.paysimple.com/3.00/paysimpleapi.svc?WSDL and the service endpoints reside at https://sandbox-api.paysimple.com/3.00/paysimpleapi/soap12.

Production environment: The production WSDL for SOAP implementation resides at https://api.paysimple.com/{version}/paysimpleapi.svc?WSDL with service endpoints at https://api.paysimple.com/{version}/paysimpleapi/soap12.

## JSON

Sandbox environment: The sandbox endpoints for the JSON service are at https://sandbox-api.paysimple.com/3.00/paysimpleapi/json

Production environment: The production endpoints for the JSON service reside at https://api.paysimple.com/{version}/paysimpleapi/json.

## XML

Sandbox environment: The sandbox endpoints for the XML service reside at https://sandbox-api.paysimple.com/3.00/paysimpleapi/xml.

Production environment: The production endpoints for the XML service reside at https://api.paysimple.com/{version}/paysimpleapi/xml

## Code Samples

Code samples may be found at http://sandbox-api.paysimple.com/3.00/help/default.html. One SOAP service is provided. All JSON and XML methods are provided and should be referred to during development to ensure correct typing of the calls. Individual url's for XML and JSON examples are:

https://sandbox-api.paysimple.com/3.00/help/XmlLog.txt
https://sandbox-api.paysimple.com/3.00/help/XmlResponseLog.txt
https://sandbox-api.paysimple.com/3.00/help/JsonLog.txt
https://sandbox-api.paysimple.com/3.00/help/JsonResponseLog.txt

## Getting Started

Merchants and their developers who wish to integrate with the PaySimple API should contact PaySimple at 1.800.466.0992 to acquire their sandbox API keys and set up an account.

Generic objects are used for each API call to create a consistent environment for development and are defined below. A userkey and an apikey provide the security and authentication layers. They are described in detail in the Security and Authentication section.

# Connecting to the PaySimple API

## SOAP

The API supports SOAP 1.2.  Most frameworks will generate your class for you.

## Rest Services

### JSON

#### *GET*

A good first test will be the call to GetStates(), at
https://sandbox-api.paysimple.com/3.00/paysimpleapi/json/getstates

This is a “GET” method. In order to properly make this request you must first create a web connection in your language. Before sending the request the headers need to be adjusted appropriately. The proper headers for the json service are:

```
Host: api.paysimple.com
Content-Type: application/json; charset=utf-8
Accept: application/json
User-Agent: CompanyName
Content-Length: 0
```

#### *POST*

A good test for a “POST” operation is GetSupportedPaymentTypes() at
https://sandbox-api.paysimple.com/3.00/paysimpleapi/json/getsupportedpaymenttypes

The headers for this request are the same, except the method is “POST” instead of get, and the Content-Length header should reflect the length of the content. The request content itself needs to be set for this post operation, such as:

```
Content-Type: application/json; charset=utf-8
Accept: application/json
User-Agent: CompanyName
Host: api.paysimple.com
Content-Length: 39

{"userkey":"USERKEY","apikey":"APIKEY"}
```

### XML

#### *GET*

A good first test will be the call to GetStates(), at
https://sandbox-api.paysimple.com/3.00/paysimpleapi/xml/getstates

This is a “GET” method. In order to properly make this request you must first create a web connection in your language. Before sending the request the headers need to be adjusted appropriately. The proper headers for the xml service are:

```
Host: api.paysimple.com
Content-Type: application/xml; charset=utf-8
Accept: application/xml
User-Agent: CompanyName
Content-Length: 0
```

### *POST*

A good test for a “POST” operation is GetSupportedPaymentTypes() at
https://sandbox-api.paysimple.com/3.00/paysimpleapi/xml/getsupportedpaymenttypes

The headers for this request are the same, except the method is “POST” instead of get, and the Content-Length header should reflect the length of the content. The request content itself needs to be set for this post operation, such as:

```
Content-Type: application/xml; charset=utf-8
Accept: application/xml
User-Agent: CompanyName
Host: api.paysimple.com
Content-Length: 135

<GetSupportedPaymentTypes xmlns="http://api.paysimple.com">
   <userkey>USERKEY</userkey>
   <apikey>APIKEY</apikey>
</GetSupportedPaymentTypes>
```

# Generic Objects

The objects used by the PaySimple API generally take into account the action of processing payments only. Extended objects may be used for reporting purposes. As much validation as possible should be declared within the objects themselves via validator attributes.

## PsResponse

The PsResponse object is the basic object returned by all PaySimple API methods requiring a userkey and apikey. This object contains a Boolean value indicating the success or failure of the method operation (note that this will not necessarily indicate the success or failure of the action, a declined payment will result in a PsResponse object indicating success, but the payment object itself may be failed).

In addition, the PsResponse object may include one or more PsObjects which are the base objects of all other API objects. The inclusion of this PsObject is determined by the success or failure indicated by the associated PsResponse object. In addition, the PsResponse object supports paging operations.

The PsResponse object is defined as follows in C#

```
public class PsResponse
{
   public bool IsSuccess { get; set; }
   public String[] ErrorMessage { get; set; }
   public string SubType { get; set; }
   public PsObject[] PsObject { get; set; }
   public int TotalItems { get; set; }
   public int ItemsPerPage { get; set; }
   public int CurrentPage { get; set; }
   public ErrorCodeType ErrorType { get; set; }
}
```

### PsResponse Member Definitions

| | |
|---|---|
| IsSuccess | Boolean value representing the success of the method call |
| ErrorMessage[] | In the case of the IsSuccess == false, the ErrorMessage will contain an array of strings indicating the reason(s) for the method failure |
| SubType | The string representation of the type of PsObject which the PsResponse contains |
| PsObject[] | Array of PsObjects returned upon a successful method call |
| TotalItems | When paging is used, the total number of items available for the method |
| ItemsPerPage | When paging is used, the number of items returned per page |
| CurrentPage | When paging is used, the value of the current page, by which the number of subsequent calls can be deduced |

| | |
|---|---|
| ErrorCodeType | The type of the error code in the case where IsSuccess == false |

## PsObject

Most api objects are subclassed from PsObject.

```
public class PsObject
{
   public int PsReferenceId { get; set; }
   public string ApiConsumerData { get; set; }
}
```

### PsObject Member Definitions

| | |
|---|---|
| ApiConsumerData | All PsObjects may pass this value in as a keyed reference to the consumer application’s datastore, maxlength = 50 |
| PsReferenceId | This is the key of the object in PaySimple’s datastore |

## PsPagedCriteria

The PsPagedCriteria object allows the user to change the result size of list operations, and request a specific page. The PsResponse object will be populated with the total number of items available and the current page.

```
public class PsPagedCriteria
{
   public int Page { get; set; }
   public int ItemsPerPage { get; set; }
}
```

### PsPagedCriteria Member Definitions

| | |
|---|---|
| Page | An integer representing what page the user wants |
| ItemsPerPage | An integer from 10 to 200 used to adjust the max number of items in the list. |

## PsCustomerPagedCriteria

The PsCustomerPagedCriteria object allows the user to change the result size of list operations, and request a specific page. In addition sorting is available via the Sort enumeration, along with column selection for sorting via the SortingSubject field. Valid sorting columns are “Id” (default), “FirstName”, “LastName”, or “CompanyName”. The PsResponse object will be populated with the total number of items available and the current page.

```
public class PsCustomerPagedCriteria
{
   public int Page { get; set; }
   public int ItemsPerPage { get; set; }
   public Sort AscOrDesc { get; set; }
   public string SortingSubject { get; set; }
}
```

### PsCustomerPagedCriteria Member Definitions

| | |
|---|---|
| Page | An integer representing what page the user wants |
| ItemsPerPage | An integer from 10 to 200 used to adjust the max number of items in the list. |
| AscOrDesc | Ascending or descending sort direction |
| SortingSubject | “Id” (default), “FirstName”, “LastName”, “CompanyName”. Values are not case sensitive |

## PsPaymentPagedCriteria

The PsPaymentPagedCriteria object allows the user to change the result size of list operations, and request a specific page. In addition sorting is available via the Sort enumeration, and the SortingSubject field is available for future use. The PsResponse object will be populated with the total number of items available and the current page.

```
public class PsPaymentPagedCriteria
{
   public int Page { get; set; }
   public int ItemsPerPage { get; set;
   public Sort AscOrDesc { get; set;
   public string SortingSubject { get; set;
   public DateTime StartDate { get; set; }
   public DateTime EndDate { get; set; }
   public PsPaymentDateRangeType DateType {get; set;}
   public int CustomerId { get; set; }
   public PsPaymentStatusFilter Filter { get; set; }
}
```

### PsCustomerPagedCriteria Member Definitions

| | |
|---|---|
| Page | An integer representing what page the user wants |
| ItemsPerPage | An integer from 10 to 200 used to adjust the max number of items in the list. |
| AscOrDesc | Ascending or descending sort direction |
| SortingSubject | “Id” (default), “FirstName”, “LastName”, |

| | |
|---|---|
| | “CompanyName”. Values are not case sensitive |
| StartDate | Start date for the date range of the search |
| EndDate | Optional end date for the date range of the search |
| PsPaymentDateRangeType | PaymentDate (0) or SettlementDate (1) |
| SortingSubject | “Id” |
| CustomerId | 0 for all payments, or a valid customer id for just that customer’s payments |
| PsPaymentStatusFilter | Object containing Boolean fields to narrow the search results based on payment status |

## PsCustomer

The PsCustomer object contains customer data for basic Create, Read, Update, and Delete (CRUD) operations, and for api method state reference. The definition and representation is as follows

```
public class PsCustomer : PsObject
{
  public string FirstName { get; set; }
  public string MiddleName { get; set; }
  public string LastName { get; set; }
  public string Email { get; set; }
  public string AltEmail { get; set; }
  public string Phone { get; set; }
  public string AltPhone { get; set; }
  public string Fax { get; set; }
  public string WebSite { get; set; }
  public string BillingAddress1 { get; set; }
  public string BillingAddress2 { get; set; }
  public string BillingCity { get; set; }
  public int BillingState { get; set; }
  public string BillingPostalCode { get; set; }
  public string BillingCountryCode { get; set; }
  public bool ShippingSameAsBilling { get; set; }
  public string ShippingAddress1 { get; set; }
  public string ShippingAddress2 { get; set; }
  public string ShippingCity { get; set; }
  public int ShippingState { get; set; }
  public string ShippingPostalCode { get; set; }
  public string ShippingCountryCode { get; set; }
  public string CompanyName { get; set; }
  public string Notes { get; set; }
  public DateTime LastModified { get; set; }
  public DateTime CreatedOn { get; set; }
}
```

### PsCustomer Member Definitions

| | |
|---|---|
| FirstName | Customer first name maxLength=150, Required |
| MiddleName | Customer middle name maxLength=150 |
| LastName | Customer last name maxLength=150, Required |

| | |
|---|---|
| Email | Customer email address maxLength=100 |
| AltEmail | Customer alternate email address maxLength=100 |
| Phone | Customer phone number maxLength=10, numeric digits only |
| AltPhone | Customer alternate phone number maxLength=10, numeric digits only |
| Fax | Customer fax number maxLenth=10, numeric digits only |
| Website | Customer website address maxLength=100 |
| BillingAddress1 | Customer billing address line 1 maxLength=250, Required |
| BillingAddress2 | Customer billing address line 2 maxLength=250 |
| BillingCity | Customer billing city maxLength=100, Required |
| BillingState | Customer billing state or province id |
| BillingPostalCode | Customer billing postal code, maxLength=10, Required |
| BillingCountryCode | Customer country code, maxLength=3 defaults to USA |
| ShippingSameAsBilling | Use this value to indicate that shipping fields should be duplicated as billing fields |
| ShippingAddress1 | Customer shipping address line 1 maxLength=250, All shipping fields default to corresponding billing fields if ShippingSameAsBililng is set to true. |
| ShippingAddress2 | Customer shipping address line 2 maxLength=250, All shipping fields default to corresponding billing fields if ShippingSameAsBililng is set to true. |
| ShippingCity | Customer shipping city maxLength=100, All shipping fields default to corresponding billing fields if ShippingSameAsBililng is set to true. |
| ShippingState | Customer shipping state or province, maxLength=2, All shipping fields default to corresponding billing fields if ShippingSameAsBililng is set to true. |
| ShippingPostalCode | Customer shipping postal code, maxLength=10, All shipping fields default to corresponding billing fields if ShippingSameAsBililng is set to true. |
| ShippingCountryCode | Customer country code, maxLength=3, All shipping fields default to corresponding billing fields if ShippingSameAsBililng is set to true. |
| CompanyName | Customer company name maxlength=50 |
| Notes | Customer additional notes maxLength=2048 |
| LastModified | DateTime UTC representation of when the customer was last modified in the PaySimple system, Read-Only |
| CreatedOn | DateTime UTC representation of when the customer was first created in the PaySimple system, Read-Only |

## PsCustomerAccount

PsCustomerAccount is the base object for PsCreditCardAccount and PsAchAccount objects.

```
public class PsCustomerAccount : PsObject
{
  public int CustomerId { get; set; }
}
```

### PsCustomerAccount Member Definitions

| | |
|---|---|
| CustomerId | PaySimple CustomerId, Required |

## PsAchAccount

PsAchAccount includes those properties specific to Ach Payments. This object extends PsCustomerAccount.

```
public class PsAchAccount : PsCustomerAccount
{
  public bool IsCheckingAccount { get; set; }
  public string RoutingNumber { get; set; }
  public string AccountNumber  { get; set; }
  public string BankName { get; set; }
}
```

### PsAchAccount Member Definitions

| | |
|---|---|
| IsCheckingAccount | If set to true, this account will be marked as checking for ach transactions, otherwise it will be marked as a savings account. |
| RoutingNumber | The routing number of the banking institution where the account resides, maxLength=9 |
| AccountNumber | The account number, maxLength = 50 |
| BankName | The name of the institution at which the account resides, maxLength = 100 |

## PsCreditCardAccount

PsCustomerCreditCardAccount contains those properties specific to credit card accounts. This object extends PsCustomerAccount.

```
public class PsCreditCardAccount : PsCustomerAccount
{
  public string CCExpiry { get; set; }
  public CreditCardIssuer CCType { get; set; }
```

```
    public string AccountNumber { get; set; }
  }
```

### PsCreditCardAccount Member Definitions

| | |
|---|---|
| CCExpiry | Credit card account expiration date. Format is MM/YYYY (e.g. 08/2010) |
| CCType | Credit Card Issuer enumeration |
| AccountNumber | The credit card account number |

## PsPayment

The PsPayment object contains the necessary information to make a payment.

```
  public class PsPayment : PsObject
  {
    public int RefPaymentId { get; set; }
    public int CustomerId { get; set; }
    public int CustomerAccountId { get; set; }
    public decimal Amount { get; set; }
    public PaymentStatus Status { get; set; }
    public DateTime PaymentDate { get; set; }
    public int InvoiceId { get; set; }
    public int RecurringScheduleId { get; set; }
    public bool IsDebit { get; set; }
    public PaymentType PaymentType { get; set; }
    public string PaymentSubType { get; set; }
    public string ProviderAuthCode { get; set; }
    public string TraceNumber { get; set; }
    public DateTime EstimateSettledDate { get; set; }
    public DateTime ActualSettledDate { get; set; }
    public DateTime CanVoidUntil { get; set; }
    public string InvoiceNumber{ get; set; }
    public string PurchaseOrderNumber { get; set; }
    public string OrderId { get; set; }
    public string Description { get; set; }
    public IDictionary<String, String> CustomData{ get; set; }
  }
```

### PsPayment Member Definitions

| | |
|---|---|
| RefPaymentId | If this payment is part of a return, then the RefPaymentId is populated with the PsReferenceId of either the original payment or the return payment |
| CustomerId | The PsReferenceId of the PsCustomer for which the payment is made |
| CustomerAccountId | The PsReferenceId of the PsCustomerAccount used for this payment |

| | |
|---|---|
| Amount | The amount of the payment |
| Status | Readonly PaymentStatus enumeration |
| PaymentDate | The date the payment was made |
| InvoiceId | If the payment references a PsInvoice, then the PsReferenceId of the invoice will appear here |
| RecurringScheduleId | If the payment was made from a PsRecurringPayment schedule, then the PsReferenceId of the payment will appear here |
| IsDebit | If this is a refund payment this field will be set to true |
| PaymentType | Readonly - CC or ACH enumeration |
| PaymentSubType | Readonly - the SubPaymentType |
| ProviderAuthCode | Readonly – the payment provider's response<br>**NOTE: This field may change at any time and is merely a pass through by PaySimple.** |
| TraceNumber | Readonly – In the case of a successful payment, the tracking field for the payment. In the case of a failed payment, the provider's error code |
| EstimateSettledDate | Readonly - The estimated date the merchant will receive payment |
| ActualSettledDate | Readonly - The actual date PaySimple received confirmation of the payment |
| CanVoidUntil | Readonly – The latest time an authorized transaction can be voided |
| InvoiceNumber | Optional field, maxLength=50 |
| PurchaseOrderNumber | Optional field, maxLength=50 |
| OrderId | Optional field, maxLength=50 |
| Description | Optional field, maxLength=2048 |
| CustomData | The customdata dictionary, if any, for this payment |

## PsPaymentOrderDetail

The PsPaymentOrderDetail object contains optional payment information for your use. Note: the variable names are not contextual, only suggestive. You may place any data you like here; PaySimple will not perform logic operations on these fields.

```
public class PsPaymentOrderDetail
{
   public string InvoiceNumber { get; set; }
   public string PurchaseOrderNumber{ get; set; }
   public string OrderId{ get; set; }
   public string Description{ get; set; }
   public IDictionary<String, String> CustomData{ get; set; }
}
```

### PsRecurringPayment Member Definitions

| | |
|---|---|
| InvoiceNumber | An optional invoice number, maxLength=50 |
| PurchaseOrderNumber | An optional purchase order number, |

| | |
|---|---|
| | maxLength=50 |
| OrderId | An optional order id, maxLength=50 |
| Description | An optional description, maxLength=2048 |
| CustomData | Optional string dictionary for your use |

### PsRecurringPayment

The PsRecurringPayment object contains the necessary information to set up and report on a recurring payment schedule. Many of the fields in recurring payments should be used for reporting purposes only.

```
public class PsRecurringPayment : PsObject
{
  public int CustomerId { get; set; }
  public int CustomerAccountId { get; set; }
  public RecurringScheduleType RecurringScheduleType { get; set; }
  public DateTime StartDate { get; set; }
  public bool HasEndDate { get; set; }
  public DateTime EndDate { get; set; }
  public BillingFrequencyType BillingFrequencyType { get; set; }
  public int BillingFrequencyParam { get; set; }
  public double PaymentAmount { get; set; }
  public bool FirstPaymentDone { get; set; }
  public double FirstPaymentAmount { get; set; }
  public DateTime FirstPaymentDate { get; set; }
  public double TotalDueAmount { get; set; }
  public int TotalNumberOfPayments { get; set; }
  public double BalanceRemaining { get; set; }
  public int NumberOfPaymentsRemaining { get; set; }
  public string InvoiceNo { get; set; }
  public string OrderId { get; set; }
  public string Description { get; set; }
  public int ScheduleStatus { get; set; }
  public int NumberOfPaymentMade { get; set; }
  public double TotalAmountPaid { get; set; }
  public DateTime DateOfLastPaymentMade { get; set; }
  public DateTime PauseUntilDate { get; set; }
}
```

### PsRecurringPayment Member Definitions

| | |
|---|---|
| CustomerId | The PsReferenceId of the customer for whom the schedule will occur, Required |
| CustomerAccountId | The PsReferenceId of the PsCustomerAccount  to be used for this payment schedule, Required |
| RecurringScheduleType | The type of schedule, from the RecurringScheduleType enumeration, Required |
| StartDate | The date the schedule will start, Required |
| HasEndDate | If the schedule has an end date, set this value to true and supply the EndDate field, otherwise set this to false. |

| | |
|---|---|
| EndDate | The optional end date of a recurring schedule |
| BillingFrequencyType | The frequency of the schedule, from the BillingFrequencyType enumeration, Required |
| BillingFrequencyParam | Optional parameter used in conjunction with BillingFrequencyType (e.g specifies the day of month on a SpecificDayOfMonth schedule) |
| PaymentAmount | The amount of the recurring payment |
| FirstPaymentDone | Boolean indicator for FirstPaymentDone |
| FirstPaymentAmount | The amount of the optional first payment |
| FirstPaymentDate | The date of the first payment |
| TotalDueAmount | The amount The total due on a payment plan schedule |
| TotalNumberOfPayments | The total number of payments due for a payment plan schedule |
| BalanceRemaining | The balance remaining for a payment plan |
| NumberOfPaymentsRemaining | The number of payments remaining for a payment plan |
| InvoiceNo | Optional InvoiceId, maxLength = 50 |
| OrderId | Optional OrderId, maxLength = 50 |
| Description | Optional description, maxLength=2048 |
| ScheduleStatus | The status of the schedule (paused, active) |
| NumberOfPaymentsMade | The number of payments made for the schedule |
| TotalAmountPaid | The total amount of payments made for the schedule |
| DateOfLastPaymentMade | The date of the last payment made for the schedule |
| PauseUntilDate | The date of schedule reactivation |

## PsRecurringPaymentFilter

The PsRecurringPaymentFilter object is used for filtering payments in the ListRecurringPayments method. Set the proper field to true to return schedules of that status.

```
public class PsRecurringPaymentFilter
{
   public bool Active { get; set; }
   public bool Disabled { get; set; }
   public bool Paused { get; set; }
   public bool Expired { get; set; }
   public bool Suspended { get; set; }
}
```

## PsPaymentStatusFilter

The PsPaymentStatusFilter object is used for filtering payments in the ListPaymentsEx method. Set the proper field to true to return payments of that status.

```
public class PsPaymentStatusFilter
{
   public bool All {get;set;}
   public bool Pending {get;set;}
   public bool Posted {get;set;}
   public bool Settled {get;set;}
   public bool Failed {get;set;}
   public bool Resubmitted {get;set;}
   public bool Voided {get;set;}
   public bool Reversed {get;set;}
   public bool Saved {get;set;}
   public bool Scheduled {get;set;}
   public bool ReversePosted {get;set;}
   public bool ChargeBack {get;set;}
   public bool CloseChargeBack {get;set;}
   public bool Authorized {get; set;}
   public bool Returned {get;set;}
   public bool ReverseChargeBack {get;set;}
   public bool ReverseNSF {get;set;}
   public bool ReverseReturn {get;set;}
   public bool RefundSettled {get;set;}
}
```

## PsInvoice

The PsInvoice object contains data specific to a PaySimple invoice.  Invoices are made up of a collection of 1 or more PsInvoiceLineItems associated with 0 - 2 PsInvoiceLineItemTaxes per PsInvoiceLineItem. PsInvoiceLineItem contains a reference to an existing PsLineItem along with an associated quantity which must be 1 or greater. PsInvoiceLineItemTax operates under a similar principle to PsInvoiceLineItems. PsInvoiceLineItemTax contains a reference to an existing PsLineItemTax.  Before sending an invoice you must have at least 1 existing PsLineItem to reference in your invoice via the PsInvoiceLineItem object.  Taxes are assigned per line item and you are allowed 2 (per line item), but taxes are optional.  There must be existing PsLineItemTax(es) to be referenced for an invoice line item in PsInvoiceLineItemTaxes.

Invoice schedules allow the repeated sending of an invoice on a specified schedule.  The invoice is created the same way as a regular invoice, but with some additional parameters.  Invoice Schedules require a BillingFrequencyType enumeration that specifies how frequently the invoice is sent out (weekly, yearly, etc.).  BillingFrequencyParam is an optional parameter that applies to certain BillingFrequencyTypes. For example, for a weekly invoice schedule BillingFrequencyParam allows the specification of which day of the week (Sunday = 1) the invoice is sent.

```
public class PsInvoice : InvoiceBase
{
   public bool IsCancelled { get; set; }
   public PsInvoiceLineItem[] InvoiceLineItems {get; set;}
   public string InvoiceAccountNumber { get; set; }
   public decimal InvoiceAmount { get; set; }
   public string InvoiceNo { get; set; }
   public DateTime DueDate { get; set; }
   public string PurchaseOrderNumber { get; set; }
   public string Description { get; set; }
   public bool IsPaid { get; set; }
   public int PaymentTransactionID { get; set; }
   public DateTime PaymentDate { get; set; }
   public DateTime LastResentOn { get; set; }
   public int ResentCount { get; set; }
   public int CustomerId { get; set; }
   public double AmountPaid { get; set; }
}
```

### PsInvoice Member Definitions

| | |
|---|---|
| IsCancelled | Boolean value indicating whether or not this invoice is still active |
| InvoiceLineItems[] | The PsInvoiceLineItem array for this invoice |
| InvoiceAccountNumber | Optional AccountNumber for the invoice, maxLength = 48 |
| InvoiceAmount | The amount due for the invoice |
| InvoiceNo | The invoice number for this invoice |
| DueDate | The date the invoice is due |
| PurchaseOrderNumber | Optional purchase order number for the invoice, maxLength = 50 |
| Description | Optional description for the invoice, maxLength = 2048 |
| IsPaid | Boolean value indicating whether or not this invoice has been paid |
| PaymentTransactionId | ThePsReferenceId of the PsPayment used to pay this invoice |
| PaymentDate | The date the invoice was paid |
| LastResentOn | The last time the invoice was sent |
| ResentCount | The number of times the invoice has been sent |
| CustomerId | The PsReferenceId of the PsCustomer for whom the invoice is for |
| AmountPaid | The amount of the invoice that has been paid |

## PsInvoiceEx

The PsInvoiceEx object contains all the same fields except for the additional fields: InvoiceFirstSentDate, InvoiceGeneratedFromSchedule, DiscountPercentage, DiscountDollarAmount, and TaxTotalAmount. Additionally, InvoiceLineItems is of type PsInvoiceLineItemEx instead of PsInvoiceLineItem. The fields InvoiceFirstSentDate, InvoiceGeneratedFromSchedule, DiscountDollarAmount and TaxTotalAmount are only used to provide data back. Any values sent in these fields will be ignored. These fields will be filled in when querying an invoice using GetInvoiceByIdEx.

```
public class PsInvoiceEx : PsInvoice
{
  public PsInvoiceLineItemEx[] InvoiceLineItems { get; set;}
  public DateTime InvoiceFirstSentDate { get; }
  public bool InvoiceGeneratedFromSchedule { get; }
  public decimal DiscountPercentage { get; set; }
  public decimal DiscountDollarAmount { get; }
  public decimal TaxTotalAmount { get; }
}
```

### PsInvoiceEx Member Definitions

| | |
|---|---|
| InvoiceLineItems[] | The PsInvoiceLineItemEx array for this invoice |
| InvoiceFirstSentDate | The date that the invoice was first sent (read only) |
| InvoiceGeneratedFromSchedule | Flag indicating whether invoice was generated from a recurring invoice schedule (read only) |
| DiscountPercentage | The percentage amount to discount the invoice amount |
| DiscountDollarAmount | Total amount of discount in dollars (read only) |
| TaxTotalAmount | Total amount of taxes in dollars (read only) |

## PsInvoiceLineItem

The PsInvoiceLineItem object contains information for specific line items within a PaySimple invoice.

```
public class PsInvoiceLineItem : PsObject
{
  public int LineItemId { get; set; }
  public decimal Quantity { get; set; }
  public PsInvoiceLineItemTax[] Taxes { get; set; }
}
```

### PsInvoiceLineItem Member Definitions

| | |
|---|---|
| LineItemId | The PsReferenceId of the PsLineItem for the invoice |
| Quantity | The quantity of the PsLineItem |
| Taxes[] | The array of taxes for the PsInvoiceLineItem |

## PsInvoiceLineItemEx

The PsInvoiceLineItemEx object contains information for specific line items within a PaySimple invoice.

```
public class PsInvoiceLineItemEx : PsInvoiceLineItem
{
   public string ItemDescription { get; set; }
   public decimal? ItemPrice { get; set; }
}
```

### PsInvoiceLineItemEx Member Definitions

| | |
|---|---|
| ItemDescription | Description of item – will override LineItemDescription from PsLineItem – if present |
| ItemPrice | Price of item – will override LineItemPrice from PsLineItem – if present |

## PsLineItem

PsLineItem is the base object of items which can appear in an invoice.

```
public class PsLineItem : PsObject
{
   public string LineItemName { get; set; }
   public string LineItemDescription { get; set; }
   public decimal LineItemPrice { get; set; }
}
```

### PsLineItem Member Definitions

| | |
|---|---|
| LineItemName | The name of the line item, maxLength=48 |
| LineItemDescription | The description for the line item, maxLength = 500 |
| LineItemPrice | The price for 1 unit of the line item |

## PsInvoiceLineitemTax

The PsInvoiceLineItemTax object contains individual taxes for line items.

```
public class PsInvoiceLineItemTax : PsObject
{
   public int TaxId { get; set; }
}
```

### PsInvoiceLineItemTax Member Definitions

| | |
|---|---|
| TaxId | The PsReferenceId of the PsLineItemTax to include for the PsInvoiceLineItem |

## PsLineItemTax

The PsLineItemTax contains the fields for taxing PsLineItems.

```
public class PsLineItemTax : PsObject
{
   public string TaxName { get; set; }
   public decimal TaxValue { get; set; }
   public bool TaxIsPercentage { get; set; }
}
```

### PsLineItemTax Member Definitions

| | |
|---|---|
| TaxName | The name of the tax |
| TaxValue | The value of the tax |
| TaxIsPercentage | True if the tax is a percentage |

## PsDefaultCustomerAccount

The PsDefaultCustomerAccount is used in the call to GetDefaultCustomerAccount.

```
public class PsDefaultCustomerAccount
{
   public int PsReferenceId { get; set; }
   public bool IsCreditCard { get; set; }
   public int CustomerId { get; set; }
}
```

### PsDefaultCustomerAccount Member Definitions

| | |
|---|---|
| IsCreditCard | Flag indicating account is a credit card account |
| CustomerId | Id number of customer account belongs to |

## PsUser

The PsUser is used in the call to GetUserInfo.

```
public class PsUser
{
   public int PsReferenceId { get; set; }
   public string FirstName { get; set; }
   public string LastName { get; set; }
   public string Company { get; set;}
   public string Email { get; set; }
   public string Phone { get; set; }
}
```

### PsUser Member Definitions

| | |
|---|---|
| FirstName | First name of user |
| LastName | Last name of user |
| Company | User’s company name if any |
| Email | Email address of user |
| Phone | Phone number of user |

## Enumerations

The PaySimple API exposes enumerations in the WSDL that should be used by the developer.

### CreditCardIssuer Enumeration

```
public enum CreditCardIssuer : int
{
   Visa = 1,
   Master = 2,
   Amex = 3,
   Discover = 4
}
```

### BillingFrequencyType Enumeration

```
public enum BillingFrequencyType : int
{
   Daily = 1,
   Weekly = 2,
   BiWeekly = 3,
   FirstofMonth = 4,
   SpecificDayofMonth = 5,
   LastofMonth = 6,
   Quarterly = 7,
   SemiAnnually = 8,
   Annually = 9
}
```

### RecurringScheduleType Enumeration

```
public enum RecurringScheduleType : int
{
   PaymentPlan = 1,
   BillPayment = 2
}
```

### ErrorCodeType

See section Error Messaging and Messages for additional information.

```
public enum ErrorCodeType
{
   None = 0,
   Login,
   Validation,
   Other
}
```

## PaymentType

```
public enum PaymentType
{
  CC = 1,
  ACH = 2
}
```

## CreditCardPaymentType

```
public enum CreditCardPaymentType
{
  Moto = 11,
  Swipe = 10
}
```

## AchPaymentType

```
public enum AchPaymentType
{
  [EnumMember(Value = "Arc")]
  Arc = 3,
  [EnumMember(Value = "Web")]
  Web = 4,
  [EnumMember(Value = "Tel")]
  Tel = 5,
  [EnumMember(Value = "Ppd")]
  Ppd = 6,
  [EnumMember(Value = "Ccd")]
  Ccd = 7,
  [EnumMember(Value = "Boc")]
  Boc = 8,
  [EnumMember(Value = "Rck")]
  Rck = 9,
  [EnumMember(Value = "Achd")]
  Achd = 16
}
```

## PsInvoiceScheduleStatus

```
public enum PsInvoiceScheduleStatus
{
  [EnumMember]
  [XmlEnum("1")]
  Expired = 1,
  [EnumMember]
  [XmlEnum("2")]
  Suspended,

  [EnumMember]
  [XmlEnum("3")]
  Draft,

  [EnumMember]
  [XmlEnum("4")]
  Active
}
```

## PsInvoiceStatus

```
public enum PsInvoiceStatus
{
  [EnumMember]
  [XmlEnum("0")]
  Paid = 0,

  [EnumMember]
  [XmlEnum("1")]
  Unpaid = 1,

  [EnumMember]
  [XmlEnum("2")]
  Draft = 2,

  [EnumMember]
  [XmlEnum("3")]
  Cancelled = 3,

  [EnumMember]
  [XmlEnum("4")]
  Overdue = 4,

  [EnumMember]
  [XmlEnum("5")]
  PaidPartially = 5
}
```

## PsScheduleStatus

```
public enum PsScheduleStatus
{
  [EnumMember]
  [XmlEnum("0")]
  Disabled = 0,
  [EnumMember]
  [XmlEnum("1")]
  Active = 1,

  [EnumMember]
  [XmlEnum("2")]
  PauseUntil = 2,

  [EnumMember]
  [XmlEnum("3")]
  Expired = 3,

  [EnumMember]
  [XmlEnum("4")]
  Suspended = 4
}
```

## PaymentStatus

```
public enum PaymentStatus : int
{
  All = -1,
  Pending = 0,
  Posted = 1,
  Settled = 2,
  Failed = 3,
  Resubmitted = 4,
  Voided = 5,
  Reversed = 6,
  Saved = 7,
  Scheduled = 8,
  ReversePosted = 9,
  ChargeBack = 10,
  CloseChargeBack = 11,
  Authorized = 12,
  Returned = 13,
  ReverseChargeBack = 14,
  ReverseNSF = 15,
  ReverseReturn = 16,
  RefundSettled = 17
}
```

## PsPaymentDateRangeType

```
public enum PsPaymentDateRangeType
{
  PaymentDate = 0,
  SettledDate = 1
```

## Sort

```
public enum Sort : int
{
   ASC = 0,
   DESC = 1
}
```

## Handling the PsResponse Object

We will use the call to GetSupportedPaymentTypes to demonstrate the PsResponse object, one will be a successful call, and the other will return an invalid login. The PsResponse object will always indicate success or failure of the method call, and the IsSuccess field should be the first property checked on the object. After verifying that the call was successful you may then extract the data contained in the PsObject array.

### JSON - Success

Here is the JSON result object from a successful request:

```
  {"d":{"__type":"PsResponse:http:\/\/api.paysimple.com","CurrentPage":0,"ErrorMessage":null,"ErrorType":0,"Is
Success":true,"ItemsPerPage":0,"PsObject":[{"__type":"PsAchPaymentType:http:\/\/api.paysimple.com","ApiCons
umerData":null,"PsReferenceId":0,"PaymentTypeId":2,"AchPaymentType":6},
  …
  {"__type":"PsCreditCardPaymentType:http:\/\/api.paysimple.com","ApiConsumerData":null,"PsReferenceId":0,"
PaymentTypeId":1,"CreditCardIssuer":4,"CreditCardPaymentType":11}],"SubType":"PsPaymentType","TotalItems":
8}}
```

### XML – Success

```
<PsResponse xmlns="http://api.paysimple.com" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <CurrentPage>0</CurrentPage>
  <ErrorMessage i:nil="true"/>
  <ErrorType>None</ErrorType>
  <IsSuccess>true</IsSuccess>
  <ItemsPerPage>0</ItemsPerPage>
  <PsObject>
    <PsObject i:type="PsAchPaymentType">
      <ApiConsumerData i:nil="true"/>
      <PsReferenceId>0</PsReferenceId>
      <PaymentTypeId>ACH</PaymentTypeId>
      <AchPaymentType>Ppd</AchPaymentType>
    </PsObject>
    …
    <PsObject i:type="PsCreditCardPaymentType">
      <ApiConsumerData i:nil="true"/>
      <PsReferenceId>0</PsReferenceId>
      <PaymentTypeId>CC</PaymentTypeId>
      <CreditCardIssuer>Discover</CreditCardIssuer>
      <CreditCardPaymentType>Moto</CreditCardPaymentType>
    </PsObject>
  </PsObject>
  <SubType>PsPaymentType</SubType>
  <TotalItems>8</TotalItems>
</PsResponse>
```

### SOAP - Success

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://api.paysimple.com/IPaySimpleApi/GetSupportedPaymentTypesResponse</a:Action
>
    <a:RelatesTo>urn:uuid:0a0a6f18-e1f1-4b00-b504-2c1f44eaa3bf</a:RelatesTo>
  </s:Header>
  <s:Body>
    <GetSupportedPaymentTypesResponse xmlns="http://api.paysimple.com">
      <GetSupportedPaymentTypesResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <CurrentPage>0</CurrentPage>
        <ErrorMessage i:nil="true"/>
        <ErrorType>None</ErrorType>
        <IsSuccess>true</IsSuccess>
        <ItemsPerPage>0</ItemsPerPage>
        <PsObject>
          <PsObject i:type="PsAchPaymentType">
            <ApiConsumerData i:nil="true"/>
            <PsReferenceId>0</PsReferenceId>
            <PaymentTypeId>ACH</PaymentTypeId>
            <AchPaymentType>Ppd</AchPaymentType>
          </PsObject>
          …
          <PsObject i:type="PsCreditCardPaymentType">
            <ApiConsumerData i:nil="true"/>
            <PsReferenceId>0</PsReferenceId>
            <PaymentTypeId>CC</PaymentTypeId>
            <CreditCardIssuer>Discover</CreditCardIssuer>
            <CreditCardPaymentType>Moto</CreditCardPaymentType>
          </PsObject>
        </PsObject>
        <SubType>PsPaymentType</SubType>
        <TotalItems>8</TotalItems>
      </GetSupportedPaymentTypesResult>
    </GetSupportedPaymentTypesResponse>
  </s:Body>
</s:Envelope>
```

## Errors

### JSON – Failure

Here we can see that the IsSuccess field is false, the ErrorType is 1 (from the ErrorCodeType above this translates into a Login error), and the ErrorMessage is “Invalid user ID or password”.

```
{"d":{"__type":"PsResponse:http:\/\/api.paysimple.com","CurrentPage":0,"ErrorMessage": "Invalid user ID or
password", "ErrorType":1,"IsSuccess":false,"ItemsPerPage":0,"PsObject":null,"SubType":null,"TotalItems":0}}
```

### XML – Failure

```
<PsResponse xmlns="http://api.paysimple.com" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <CurrentPage>0</CurrentPage>
  <ErrorMessage>Invalid user ID or password</ErrorMessage>
  <ErrorType>Login</ErrorType>
  <IsSuccess>false</IsSuccess>
  <ItemsPerPage>0</ItemsPerPage>
  <PsObject i:nil="true"/>
  <SubType i:nil="true"/>
  <TotalItems>0</TotalItems>
</PsResponse>
```

### SOAP – Failure

```
  <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
    <s:Header>
      <a:Action
s:mustUnderstand="1">http://api.paysimple.com/IPaySimpleApi/GetSupportedPaymentTypesResponse</a:Action
>
      <a:RelatesTo>urn:uuid:565dd86f-24f9-4ca9-9a88-a45f3d583954</a:RelatesTo>
    </s:Header>
    <s:Body>
      <GetSupportedPaymentTypesResponse xmlns="http://api.paysimple.com">
        <GetSupportedPaymentTypesResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
          <CurrentPage>0</CurrentPage>
          <ErrorMessage>Invalid user ID or password</ErrorMessage>
          <ErrorType>Login</ErrorType>
          <IsSuccess>false</IsSuccess>
          <ItemsPerPage>0</ItemsPerPage>
          <PsObject i:nil="true"/>
          <SubType i:nil="true"/>
          <TotalItems>0</TotalItems>
        </GetSupportedPaymentTypesResult>
      </GetSupportedPaymentTypesResponse>
    </s:Body>
  </s:Envelope>
```

### Validation Errors

The following responses indicate validation errors when making a call to AddCustomer

#### JSON Validation Error

```
{"d":{"__type":"PsResponse:http:\/\/api.paysimple.com","CurrentPage":0,"ErrorMessage":["Phone must be 10
digits","Billing PostalCode must be in the format 11111, 11111-1111, or A1A 1A1","Shipping address1 is required if
shipping address is not the same as billing.","Shipping city is required if shipping address is not the same as
billing.","Shipping postal code is required if shipping address is not the same as
billing."],"ErrorType":2,"IsSuccess":false,"ItemsPerPage":0,"PsObject":null,"SubType":null,"TotalItems":0}}
```

## XML Validation Error

```
<PsResponse xmlns="http://api.paysimple.com" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <CurrentPage>0</CurrentPage>
  <ErrorMessage xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
    <a:string>Phone must be 10 digits</a:string>
    <a:string>Billing PostalCode must be in the format 11111, 11111-1111, or A1A 1A1</a:string>
    <a:string>Shipping address1 is required if shipping address is not the same as billing.</a:string>
    <a:string>Shipping city is required if shipping address is not the same as billing.</a:string>
    <a:string>Shipping postal code is required if shipping address is not the same as billing.</a:string>
  </ErrorMessage>
  <ErrorType>Validation</ErrorType>
  <IsSuccess>false</IsSuccess>
  <ItemsPerPage>0</ItemsPerPage>
  <PsObject i:nil="true"/>
  <SubType i:nil="true"/>
  <TotalItems>0</TotalItems>
</PsResponse>
```

## SOAP Validation Error

```
  <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
    <s:Header>
      <a:Action
s:mustUnderstand="1">http://api.paysimple.com/IPaySimpleApi/AddCustomerResponse</a:Action>
      <ActivityId CorrelationId="9098093e-cd3c-42fb-8ddf-ba3a2ccac314"
xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">c32ad85d-8513-467c-84c4-
ebd90d21ceb1</ActivityId>
      <a:RelatesTo>urn:uuid:f0e17c25-abc0-4d7f-8a52-9b1275fe0267</a:RelatesTo>
    </s:Header>
    <s:Body>
      <AddCustomerResponse xmlns="http://api.paysimple.com">
        <AddCustomerResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
          <CurrentPage>0</CurrentPage>
          <ErrorMessage xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
            <b:string>Phone must be 10 digits</b:string>
            <b:string>Billing PostalCode must be in the format 11111, 11111-1111, or A1A 1A1</b:string>
            <b:string>Shipping address1 is required if shipping address is not the same as billing.</b:string>
            <b:string>Shipping city is required if shipping address is not the same as billing.</b:string>
            <b:string>Shipping postal code is required if shipping address is not the same as billing.</b:string>
          </ErrorMessage>
          <ErrorType>Validation</ErrorType>
          <IsSuccess>false</IsSuccess>
          <ItemsPerPage>0</ItemsPerPage>
          <PsObject i:nil="true"/>
          <SubType i:nil="true"/>
          <TotalItems>0</TotalItems>
        </AddCustomerResult>
      </AddCustomerResponse>
    </s:Body>
  </s:Envelope>
```

## Other Errors

Business logic errors will be indicated with an error type of Other (ErrorCodeType 3). An example error is below.

### JSON Error

```
{"d":{"__type":"PsResponse:http:\/\/api.paysimple.com","CurrentPage":0,"ErrorMessage":["Customer ID does not
match any Customers of this
Client"],"ErrorType":3,"IsSuccess":false,"ItemsPerPage":0,"PsObject":null,"SubType":null,"TotalItems":0}}
```

### XML Error

```
<PsResponse xmlns="http://api.paysimple.com" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <CurrentPage>0</CurrentPage>
  <ErrorMessage xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
    <a:string>Customer ID does not match any Customers of this Client</a:string>
  </ErrorMessage>
  <ErrorType>Other</ErrorType>
  <IsSuccess>false</IsSuccess>
  <ItemsPerPage>0</ItemsPerPage>
  <PsObject i:nil="true"/>
  <SubType i:nil="true"/>
  <TotalItems>0</TotalItems>
</PsResponse>
```

## PsResponse and MakePayment

As stated previously, the IsSuccess property of the PsResponse object indicates success of the method call. In the case of making a payment the IsSuccess field **does not indicate the approval or decline of the payment itself.** Here is an XML example of this behavior. You can see that the IsSuccess property is set to true, indicating we successfully processed the payment. However the payment itself was declined.

```
<PsResponse xmlns="http://api.paysimple.com" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <CurrentPage>0</CurrentPage>
  <ErrorMessage i:nil="true"/>
  <ErrorType>None</ErrorType>
  <IsSuccess>true</IsSuccess>
  <ItemsPerPage>0</ItemsPerPage>
  <PsObject>
    <PsObject i:type="PsPayment">
      <ApiConsumerData i:nil="true"/>
      <PsReferenceId>88</PsReferenceId>
      <Amount>200</Amount>
      <CustomData i:nil="true" xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
      <CustomerAccountId>216</CustomerAccountId>
      <CustomerId>791063</CustomerId>
      <Description i:nil="true"/>
      <InvoiceId>0</InvoiceId>
      <InvoiceNumber i:nil="true"/>
      <IsDebit>false</IsDebit>
      <OrderId i:nil="true"/>
      <PaymentDate>0001-01-01T00:00:00</PaymentDate>
      <PurchaseOrderNumber i:nil="true"/>
      <RecurringScheduleId>0</RecurringScheduleId>
      <RefPaymentId>0</RefPaymentId>
      <Status>Failed</Status>
    </PsObject>
  </PsObject>
  <SubType>PsPayment</SubType>
  <TotalItems>1</TotalItems>
</PsResponse>
```

## Method Signatures

Calls to the methods supported by the PaySimple API are described below. For JSON and XML requests to methods which take userkey and apikey parameter the http verb should be "POST", for other's the verb should be "GET".

### GetStates

Returns the state enumeration.

```
List<State> GetStates()
```

### AddCustomer

Adds a customer.  AddCustomer accepts a PsCustomerObject with PsReferenceId of 0
Returns a PsCustomer object.

```
PsResponse AddCustomer(string userkey, string apikey, PsCustomer customer)
```

### GetCustomer

Returns a PsCustomer object.

```
PsResponse GetCustomer(string userkey, string apikey, int id);
```

### GetCustomerByConsumerData

Get a customer based on the api user's data

```
GetCustomerByConsumerData(string userkey, string apikey, string consumerData)
```

### GetCustomerAndDefaultAccounts

Returns a PsCustomer, PsCreditCardAccount, PsAchAccount object.

```
PsResponse GetCustomerAccountByAccountId(string userkey, string apikey, int customerId);
```

### GetCustomerAccountByAccountId

Returns a PsCustomerAccount object.

```
PsResponse GetCustomerAccountByAccountId(string userkey, string apikey, int customerId, int accountId);
```

### UpdateCustomer

Updates an existing PsCustomer and returns the updated PsCustomer.

```
PsResponse UpdateCustomer(string userkey, string apikey, PsCustomer customer);
```

### DeleteCustomer

Delete a customer.  Returns true if the operation was successful.

```
PsResponse DeleteCustomer(string userkey, string apikey, int id);
```

### ListCustomers

List customers using paging criteria.
Returns a list of PsCustomers in PsObject[]

```
PsResponse ListCustomers(string userkey, string apikey, PsPagedCriteria criteria);
```

### ListCustomersWithSort

List customer using the PsCustomerPagedCriteria

```
PsResponse ListCustomerWithSort(string userkey, string apikey, PsCustomerPagedCriteria criteria)
```

### SearchCustomer

Returns a single PsCustomer if fuzzySearch is set to false, otherwise returns all customers matching the given criteria.

```
  PsResponse SearchCustomer(string userkey, string apikey, string firstName, string lastName, string company,
bool fuzzySearch, PsPagedCriteria criteria);
```

## SearchCustomerEx

Similar to SearchCustomer but allows for apiconsumerdata search

```
  PsResponse SearchCustomer(string userkey, string apikey, string firstName, string lastName, string company,
string apiConsumerData, bool fuzzySearch, PsPagedCriteria criteria);
```

## AddCustomerAchAccount

Adds a new ACH Account for a customer.  PsResponse contains a single PsAchAccount

```
  PsResponse AddCustomerAchAccount(string userkey, string apikey, PsAchAccount customerAccount);
```

## UpdateCustomerAchAccount

UpdateCustomerAchAccount accepts an existing PsAchAccount.  PsResponse contains the updated PsAchAccount

```
  PsResponse UpdateCustomerAchAccount(string userkey, string apikey, PsAchAccount customerAccount);
```

## DeleteCustomerAchAccount

Delete the specified ACH account.  Success is indicated by the IsSuccess field of PsResponse

```
  PsResponse DeleteCustomerAchAccount(string userkey, string apikey, int accountId);
```

## AddCustomerCreditCardAccount

Add a credit card account for a customer.  PsResponse contains a single PsCustomerCreditCardAccount.

```
  PsResponse AddCustomerCreditCardAccount(string userkey, string apikey, PsCreditCardAccount
customerAccount);
```

## UpdateCustomerCreditCardAccount

UpdateCustomerCreditCardAccount accepts a PsCreditCardAccount object.  PsResponse contains a single PsCustomerCreditCardAccount.

```
  PsResponse UpdateCustomerCreditCardAccount(string userkey, string apikey, PsCreditCardAccount
customerAccount);
```

### DeleteCustomerCreditCardAccount

DeleteCustomerCreditCardAccount accepts only a customer account id. PsResponse will contain IsSuccess true or false indicating success.

```
  PsResponse DeleteCustomerCreditCardAccount(string userkey, string apikey, int accountId);
```

### MakePayment

Make a payment using the customer Account ID specified.

```
  PsResponse MakePayment(string userkey, string apikey, int customerId, int customerAccountId, decimal amount,
string cid, PsPaymentOrderDetail detail);
```

### CancelPayment

Cancel a particular payment using the Payment ID.

```
  PsResponse CancelPayment(string userkey, string apikey, int paymentId);
```

### ReversePayment

Reverse a payment using the Payment ID.

```
  PsResponse ReversePayment(string userkey, string apikey, int paymentId);
```

### AddCustomerAndMakeCCPayment

Add a customer and make a credit card payment. When successful, The PsObject[] array of PsResponse contains the PsCustomer at the first element, the PsCustomerAccount as the second element, and a PsPayment object as the third element.

```
  PsResponse AddCustomerAndMakeCCPayment(string userkey, string apikey, PsCustomer customer,
PsCreditCardAccount customerAccount, decimal amount, string cid);
```

### AddCustomerAndMakeACHPayment

Add a customer and make an ACH payment.  When successful, The PsObject[] array of PsResponse contains the PsCustomer at the first element, the PsCustomerAccount as the second element, and a PsPayment object as the third element.

```
  PsResponse AddCustomerAndMakeACHPayment(string userkey, string apikey, PsCustomer customer,
PsAchAccount customerAccount, decimal amount);
```

### ListCreditCards

Returns a list of the credit cards for a customer.

```
  PsResponse ListCreditCards(string userkey, string apikey, int customerId);
```

### ListAch

Returns a list of ACH bank accounts for a customer.

```
  PsResponse ListAch(string userkey, string apikey, int customerId);
```

### GetSupportedPaymentTypes

Returns a list of the payment types available for the merchant account.

```
  PsResponse GetSupportedPaymentTypes(string userkey, string apikey);
```

### AddRecurringPayment

Add a new recurring payment.

```
  PsResponse AddRecurringPayment(string userkey, string apikey, PsRecurringPayment recurringPayment);
```

### ModifyRecurringPaymentSchedule

Change the payment amount, number of payments, billing frequency, and billing frequency param

```
  PsResponse ModifyRecurringPaymentSchedule(string userkey, string apikey, PsRecurringPayment
paymentSchedule)
```

### DeleteRecurringSchedule

Deletes a recurring payment schedule from the system

```
  PsResponse DeleteRecurringSchedule(string userkey, string apikey, int scheduleId)
```

### SuspendRecurringPaymentSchedule

Stop a recurring schedule from processing

```
  PsResponse SuspendRecurringPaymentSchedule(string userkey, string apikey, int scheduleId)
```

### ResumeRecurringPaymentSchedule

Resume processing on a previously suspended recurring schedule

```
  PsResponse ResumeRecurringPaymentSchedule(string userkey, string apikey, int scheduleId)
```

### ListRecurringPayments

Returns a list of recurring payment schedules based on status, date, and customer.  A customer id of 0 does not filter on the customer.

```
  public PsResponse ListRecurringPayments(string userkey, string apikey, DateTime startDate, DateTime endDate,
int customerId, PsRecurringPaymentFilter filter, PsPagedCriteria criteria)
```

### AddInvoice

Add a new invoice.

```
PsResponse AddInvoice(string userkey, string apikey, PsInvoice invoice);
```

### AddInvoiceEx

Add a new PsInvoiceEx object.

```
PsResponse AddInvoiceEx(string userkey, string apikey, PsInvoiceEx invoiceEx);
```

### ListLineItems

Returns a list of the line items.

```
PsResponse ListLineItems(string userkey, string apikey);
```

### AddLineItem

Add a new line item.

```
   PsResponse AddLineItem(string userkey, string apikey, PsLineItem lineItem);
```

### UpdateLineItem

Modify an existing line item.

```
PsResponse UpdateLineItem(string userkey, string apikey, PsLineItem lineItem);
```

### DeleteLineItem

Delete a line item.

```
PsResponse DeleteLineItem(string userkey, string apikey, int lineItemId);
```

## ListTaxes

List all taxes.

```
PsResponse ListTaxes(string userkey, string apikey);
```

## AddTax

Add a new tax.

```
PsResponse AddTax(string userkey, string apikey, PsLineItemTax tax);
```

## GetInvoiceNumber

Get an invoice number

```
   PsResponse GetInvoiceNumber(string userkey, string apikey);
```

## ListScheduledInvoices

Returns a list of invoice schedules based on status, date, and customer.  A customer id of 0 does not filter on the customer.

```
  public PsResponse ListScheduledInvoices(string userkey, string apikey, DateTime startDate, DateTime endDate,
int customerId, List<PsInvoiceScheduleStatus> status, PsPagedCriteria criteria)
```

## ListInvoices

Returns a list of invoices based on status, date, and customer.  A customer id of 0 does not filter on the customer.

```
  public PsResponse ListInvoices(string userkey, string apikey, DateTime startDate, DateTime endDate, int
customerId, List<PsInvoiceStatus> status, PsPagedCriteria criteria)
```

## GetInvoiceById

Returns an invoice based on the id provided

```
public PsResponse GetInvoiceById(string userkey, string apikey, int invoiceId)
```

## GetInvoiceByIdEx

Returns a PsInvoiceEx object based on the id provided

```
public PsResponse GetInvoiceByIdEx(string userkey, string apikey, int invoiceId)
```

## ResendInvoice

Resends an existing invoice

```
public PsResponse ResendInvoice(string userkey, string apikey, int invoiceId)
```

## ListPayments

List the payments for a specific customer

```
PsResponse ListPayments(string userkey, string apikey, int customerId, PsPagedCriteria criteria);
```

## ListPaymentsEx

List payments using the PsPaymentPagedCriteria object

```
PsResponse ListPaymentsEx(string userkey, string apikey, PsPaymentPagedCriteria criteria)
```

## ListPaymentsByInvoiceId

List the payments tied to an invoice

```
PsResponse ListPaymentsByInvoiceId(string userkey, string apikey, int invoiceId, PsPagedCriteria criteria)
```

### ListPaymentsByRecurringSchedule

List all the payments made for a recurring schedule

```
ListPaymentsByRecurringSchedule(string userkey, string apikey, int scheduleId, PsPagedCriteria criteria)
```

### GetStatusesbyDate

Returns the transactions whose statuses have changed for a given date range.

```
  PsResponse GetPaymentStatus(string userkey, string apikey, DateTime startDate, DateTime endDate,
PsPagedCriteria criteria);
```

### GetRecurringPaymentSchedule

Returns a RecurringPaymentSchedule by its PsReferenceId

```
GetRecurringPaymentSchedule(string userkey, string apikey, int scheduleId);
```

### GetDefaultCustomerAccount

Returns the default account to use for a customer. The PsResponse returns a PsDefaultCustomerAccount object

```
GetDefaultCustomerAccount(string userkey, string apikey, int customerId);
```

### SetDefaultCustomerAccount

Sets the default account for a customer.

```
SetDefaultCustomerAccount(string userkey, string apikey, int customerId, int customerAccountId);
```

### GetDefaultCreditCardAccount

Gets the default credit card for a customer

```
GetDefaultCreditCardAccount(string userkey, string apikey, int customerId)
```

### GetDefaultAchAccount

Gets the default ach account for a customer

```
GetDefaultAchAccount(string userkey, string apikey, int customerId)
```

### GetUserInfo

Gets the information about the current user. Returns a PsUser object

```
GetUserInfo(string userkey, string apikey)
```

### ListAllCustomerAccounts

Returns array of PsCustomerAccount objects

```
ListAllCustomerAccounts(string userkey, string apikey, int customerId)
```

## Error Messaging and Messages

PaySimple error messages are designed to be straightforward, so that the developer may make some decision based on the exact response. Since the messages may change, we provide directional information below.

Login error messages are straightforward and mean either the userkey or the apikey are incorrect. The developer should correct the issue and retry.

Validation errors mean that the data validation failed for a given piece of data.

“Other” errors mean the business logic is incorrect.

A successful response does not mean the payment succeeded. You must extract the payment object and check the payment status to glean the status of the transaction.

Below are credit card codes available which are subject to change by processors.

| | |
|---|---|
| 01 | Refer to issuer. |
| 02 | Refer to issuer - special condition |

03 Invalid Merchant ID

04 Pick up the card, there is a problem with the card.

05 Do Not Honor

07 Pick up the card. There is a problem with the card. It has been marked for fraud. Call the 800 number on the back of the card to determine the issue.

12 Invalid Transaction

14 Invalid Card Number

15 No such issuer

19 Re-Enter transaction

28 The card holder’s bank is not replying to the credit card transaction. Try waiting and then rerunning the transaction.

41 Lost Card, pick up. Fraud has been reported on this account.

43 Stolen Card, pick up. Fraud has been reported on this account.

51 Insufficient Funds

54 Expired Card

57 The bank that has issued this card does not allow this type of purchase. Call the 800 number on the back of their card for further explanation.

61 Exceeds issuer withdrawal limit. Have the customers call the 800 number on the back of the card and get a force code. This will allow you to bypass the limit.

62 Invalid service code, restricted

63 Security Violation

65 Activity limit exceeded.

78 Account not recognized

91 The card holders bank is not replying to the credit card transaction. Try waiting and then rerunning the transaction.

100 Cardholder's bank did not approve transaction.

101 Authorizer needs more information for approval.

| | |
|---|---|
| 102 | Card issuer does not want that card used. Call for further instructions. |
| 103 | Pick up card and call issuer. |
| 104 | Authorizer didn't respond within allotted time. |
| 105 | Account #/mag stripe is invalid. |
| 106 | Expiration date is either incorrect format or prior to today. |
| 107 | The Personal ID (such as PIN for a debit transaction) is incorrect. |
| 120 | Authorization center cannot be reached. |
| 121 | Card has been reported lost or stolen. Pick up card and call issuer. |
| 130 | Amount requested exceeds available funds or credit. |
| 154 | Merchant cannot accept this card. |
| 156 | Merchant cannot accept this BIN range. |
| 175 | The batch containing the transaction to void has been released. |
| 176 | Merchant is Authorization Only and a debit record was sent. |
| 177 | Manually entered transactions are not allowed for this terminal ID. |
| 178 | Maximum sale amount exceeded. |
| 179 | Maximum return amount exceeded. |
| 180 | The batch containing the transaction has been released. |
| 181 | CVC2 or CID data entered is not correct for the card. |
| CV | Card Type Verification Error |
| N7 | CVV2 Value supplied is invalid |

ACH Return Codes may be found here:

http://www.paysimple.com/paysimple30_help/PaySimple_Help.htm

## Security and Authentication

Authentication is provided by the combination of a userkey and apikey. Both keys will be provided to you when you are approved for API development. Once certified in the PaySimple sandbox, the production keys will be provided.

PaySimple security and authentication consists of:

1. A required API PaySimple-issued Username (userkey)
2. A required API PaySimple-issued API Key (apikey)
3. Secure Sockets Layer (SSL) data transport

PaySimple is a Payment Card Industry (PCI) Level 1 Merchant. Storing credit card information requires merchants who handle credit cards to meet certain security standards. PaySimple strongly recommends that merchants and developers check out the PCI requirements before authorizing credit card payments. The PCI standards can be found at the following web address: https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml.