



## Getting started with Express Checkout integration

<b>1</b>	<b>PREFACE.....</b>	<b>3</b>
<b>2</b>	<b>PAYPAL TEST ENVIRONMENT (SANDBOX) .....</b>	<b>4</b>
2.1	SANDBOX .....	4
2.2	GENERATE API CREDENTIALS .....	6
2.3	SDK .....	8
<b>3</b>	<b>EXPRESS CHECKOUT TRANSACTION FLOW :.....</b>	<b>9</b>
3.1	TRANSACTION FLOW .....	9
3.2	EXPRESS CHECKOUT AS PAYMENT MARK .....	10
3.2.1	<i>EC with landing Page</i> :.....	11
3.2.2	<i>EC without landing Page</i> .....	12
3.3	EXPRESS CHECKOUT AS SHORTCUT.....	13
<b>4</b>	<b>EXPRESS CHECKOUT API DEFINITION .....</b>	<b>14</b>
4.1	SECURITY PARAMETERS.....	14
4.2	BASE DEFINITION OF ALL API CALLS .....	15
4.3	SETEXPRESSCHECKOUT .....	16
4.4	GETEXPRESSCHECKOUTDETAILS .....	18
4.5	DOEXPRESSCHECKOUTPAYMENT .....	19
<b>5</b>	<b>USEFUL LINKS .....</b>	<b>22</b>

# 1 Preface

The goal of this document is to help merchants understand better the integration work that needs to be done to use PayPal Express Checkout and help merchants getting started.

The focus here is on the minimum requirements to implement a PayPal payment using the Express Checkout API. Only the required fields are explained.

**This document should always be used in conjunction with the official PayPal API reference guide and general documentation**

For more detailed documentation on all function and fields that can be used, please refer to the Express Checkout documentation

The following list of document should be used with this document :

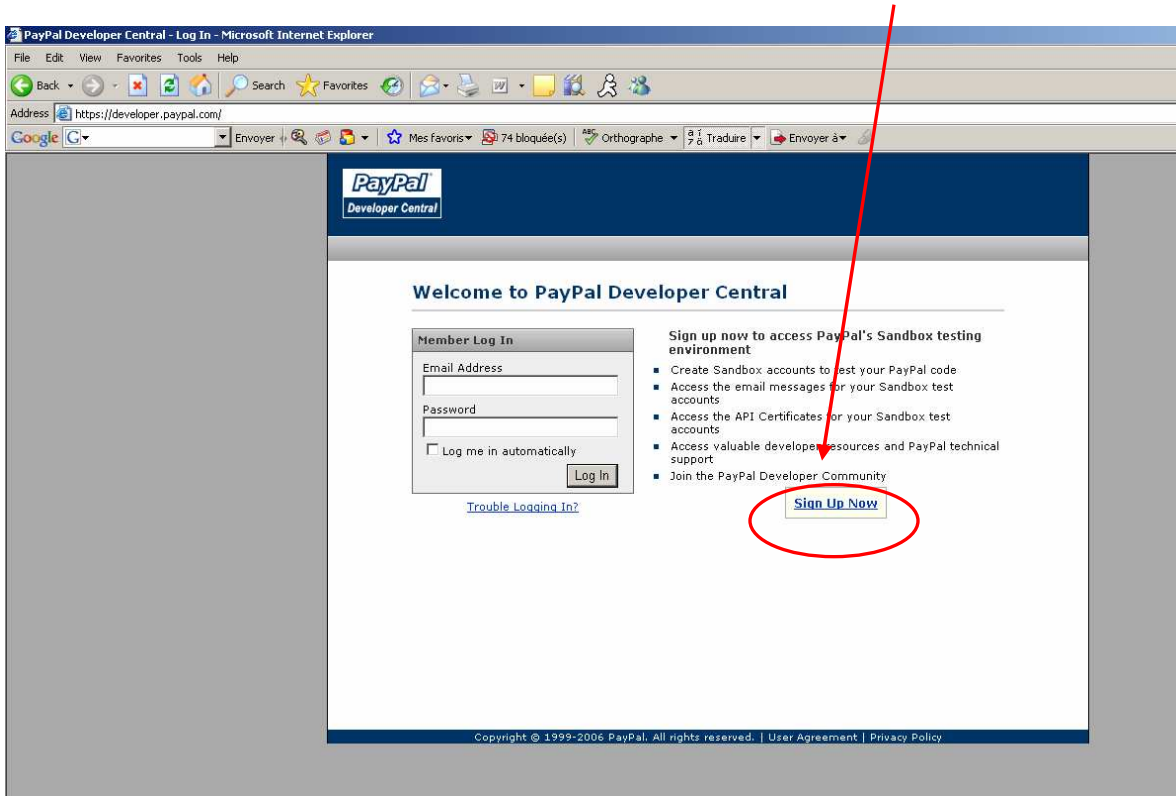
- PP\_ExpressCheckout\_IntegrationGuide.pdf
- PP\_APIReference.pdf
- PP\_Sandbox\_UserGuide.pdf
- PP\_AuthCapture\_IntegrationGuide.pdf
- PP\_NVPAPI\_DeveloperGuide.pdf

## 2 PayPal test environment (Sandbox)

### 2.1 Sandbox

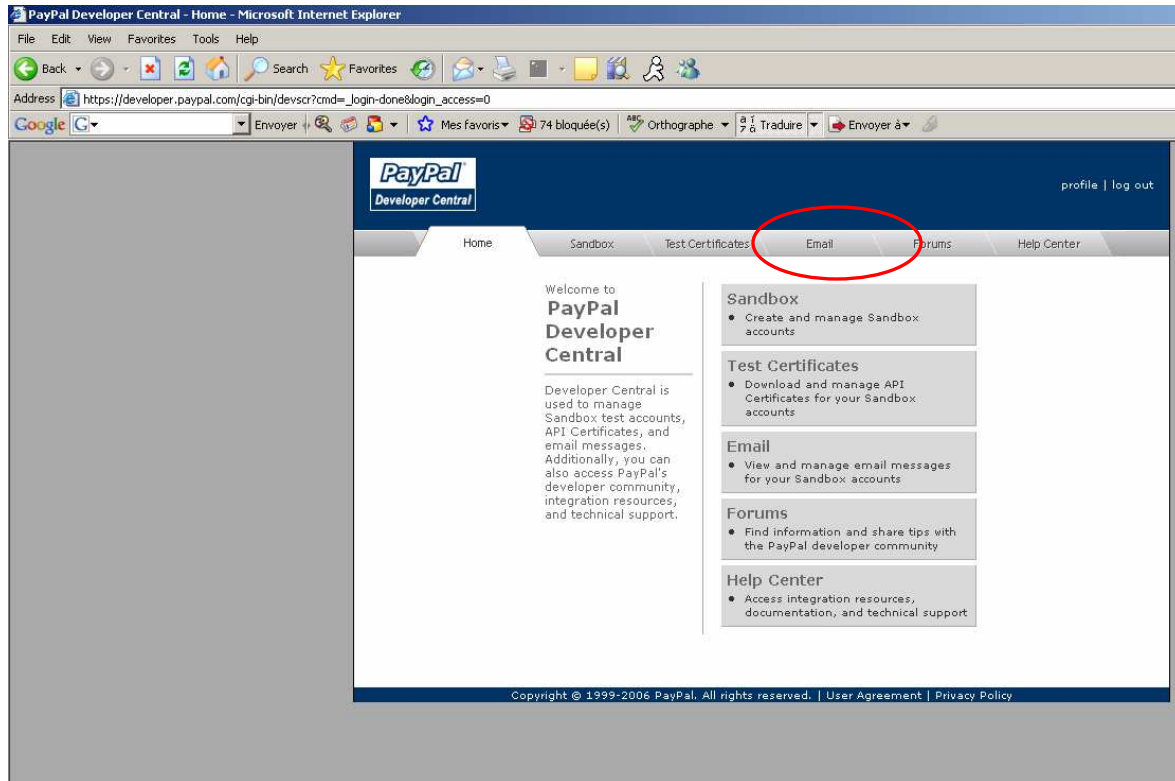
To test your PayPal integration, you can use Sandbox test accounts:

First, you need to connect to <https://developer.paypal.com> and open an account.



You will receive a email to confirm the email address you entered.

Once connected, you have access to the following items:



- **Sandbox** The sandbox allows the creation of fake PayPal account. You can create as many account as needed and the email address for the account does not to be real (**no email is sent out**).
- **Test Certificates** Download of API certificates created in PayPal account
- **Email** **All the emails sent by PayPal are in this link**

Each time you open a PayPal account on the sandbox, PayPal will sent a mail to verify the email address. As no real emails are sent in the sandbox, you **MUST** check the “**email**” link in the developer central to confirm the link.

First, create two account :

- The first account (“premier” or “business”) will act as you merchants account.
- A second account as buyer account.

When you create a account and want to add a card or a bank number, the sandbox will automatically generated numbers for you. To verify the account, start the verification process and validated it by entering any random number.

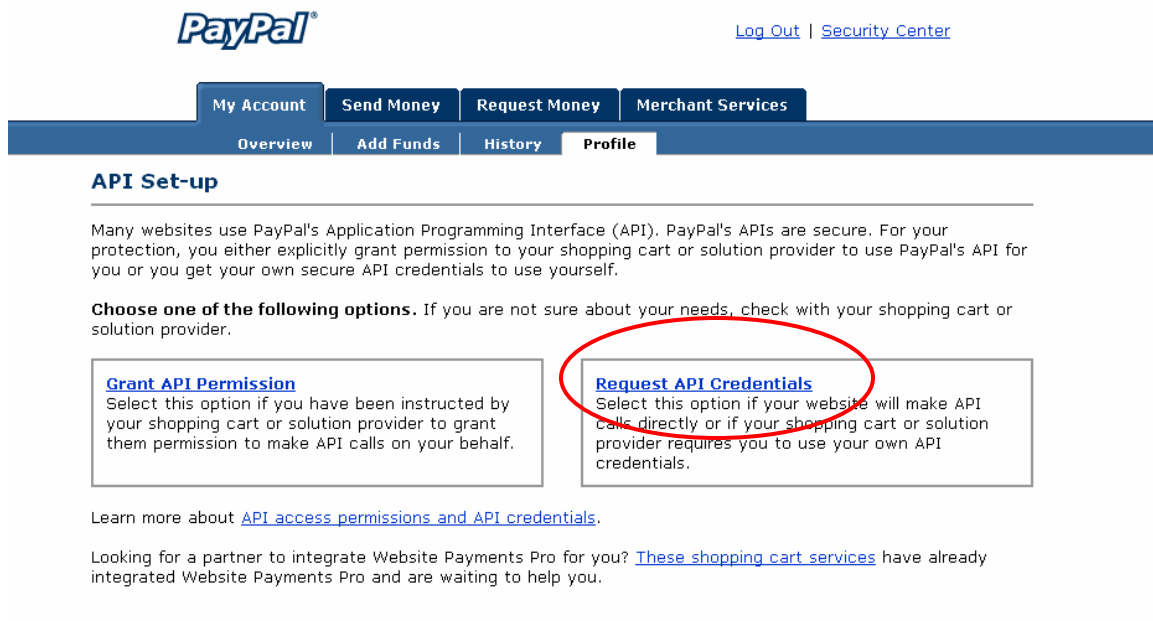
For more information, please refer to “PP\_Sandbox\_UserGuide.pdf”.

## 2.2 Generate API credentials

In order to use the PayPal Express Checkout API, you will need to generate the API credentials.

To do this, log into your PayPal account (or the Sandbox account), and choose “Profile”, “API Access”.

To request your API credentials, click on “Request API credentials”.



**API Set-up**

Many websites use PayPal's Application Programming Interface (API). PayPal's APIs are secure. For your protection, you either explicitly grant permission to your shopping cart or solution provider to use PayPal's API for you or you get your own secure API credentials to use yourself.

**Choose one of the following options.** If you are not sure about your needs, check with your shopping cart or solution provider.

**Grant API Permission**  
Select this option if you have been instructed by your shopping cart or solution provider to grant them permission to make API calls on your behalf.

**Request API Credentials**  
Select this option if your website will make API calls directly or if your shopping cart or solution provider requires you to use your own API credentials.

Learn more about [API access permissions and API credentials](#).

Looking for a partner to integrate Website Payments Pro for you? [These shopping cart services](#) have already integrated Website Payments Pro and are waiting to help you.

You can choose between two types of API credentials:

- **API Signature:** A PayPal-generated unique digital signature (a line of text, or hash) that you copy from PayPal's website and include in your API calls. An alternative to API Certificate security. Your digital signature, your API username, and your API password all together are called "three-token authentication," because you include them as programatic tokens in your API calls
- **API SSL client-side certificate:** A PayPal-generated unique digital certificate file that you download from the PayPal website and use on the client computer to encrypt the HTTPS requests of your API calls. An alternative to API signatures

## Request API Credentials

[Back to Profile Summary](#)

API credentials identify you when you make API calls. Credentials consist of an API username, an API password, and a third credential that can be either an API signature or an API SSL client-side certificate.

If you are relying on a shopping cart or integrator, ask your shopping cart or integrator which kind of credential you need.

Credential	Description
<input checked="" type="radio"/> <b>API Signature</b>	An API signature is the easiest credential to use. It is a 128-bit signature you include in your API calls to identify yourself as the authorized caller.
<input type="radio"/> <b>API SSL client-side certificate</b>	This credential may require that you be able to install the SSL certificate on your web server.

Need help deciding which credential is right for your needs? Learn more about [API credentials](#).

**Terms of Use** – The API License Agreement below protects and informs you of your rights in using PayPal APIs.

[\(Printer Friendly Version\)](#)

PayPal License Program

PayPal Inc.

API License Agreement

☒ I agree to the API Certificate Terms of Use and terms incorporated therein.



API Signature is the easiest and recommended authentication method.

## View or Remove API Signature

[Back to Profile Summary](#)

**Developers:** Do not share your credential or use it in a manner in which others can access it. Consider storing the credential as a properties file in a secure location other than your web server document root and setting the file permissions so that only the system user executing your ecommerce application can access it.

**Credential:** API Signature

**API Username:** [REDACTED]

**API Password:** BPDE45LTSY3SUP84

**Signature:** AJvDv91m2FMW8vdVBL1NZK2nnlVNA0tIpblyXx8fFNLgJavAIA85xD4z

**Request Date:** Mar. 27, 2007 01:56:58 PDT



[Mobile](#) | 
 [Money Market](#) | 
 [ATM/Debit Card](#) | 
 [Referrals](#) | 
 [About Us](#) | 
 [Accounts](#) | 
 [Fees](#) | 
 [Privacy](#) | 
 [Plus Card](#) | 
 [Security Center](#) | 
 [Contact Us](#) | 
 [Legal Agreements](#) | 
 [Developers](#)



[About SSL Certificates](#)

Copyright © 1999-2007 PayPal. All rights reserved.  
[Information about FDIC pass-through insurance](#)

## 2.3 SDK

The PayPal Express Checkout API is available using two different protocols: Soap and NVP.

Both protocols are providing SDKs in PHP, Java, ASP and .NET.

To access the SDKs using **SOAP**

[https://www.paypal.com/IntegrationCenter/ic\\_sdk-resource.html](https://www.paypal.com/IntegrationCenter/ic_sdk-resource.html)

To access the SDKs using **NVP** :

[https://www.paypal.com/IntegrationCenter/ic\\_nvp.html](https://www.paypal.com/IntegrationCenter/ic_nvp.html)

The examples on this document refer to the SOAP API. The fields might differ for the NVP API. Please refer to the corresponding guides during implementation for the correct fields and calls of the different APIs.



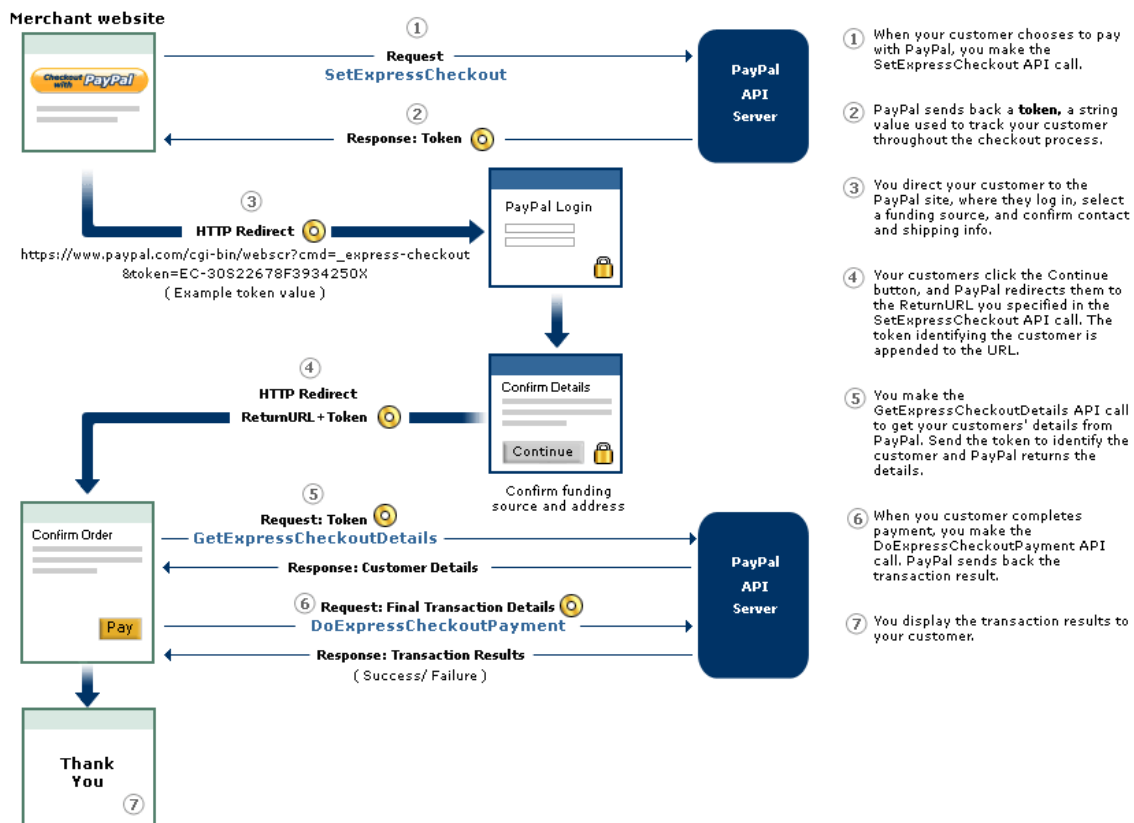
### 3 Express checkout transaction flow :

#### 3.1 Transaction flow

The following table is a brief overview of the Express Checkout payment flow:

#### PayPal Express Checkout

Legend: ➡ Web Flow ➡ API Call 🕒 Token



The following table explains the flow and the integration point :

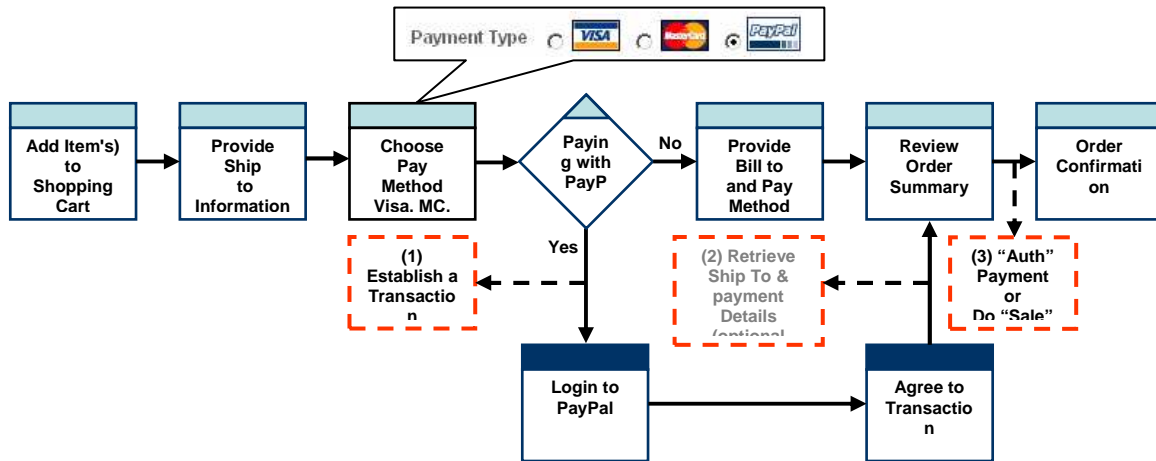
Customer	Merchant	PayPal
1 Clicks <b>Checkout with PayPal</b> button.	Calls <code>SetExpressCheckout</code> API with the required fields – estimated <code>OrderTotal</code> , <code>ReturnURL</code> , and <code>CancelURL</code> – and optional fields, such as <code>MaxAmount</code> .	
		2. Returns <code>SetExpressCheckout</code> response with <code>Token</code> and appends the values of <code>Token</code> and <code>PayerID</code> to your return URL.
	3. Adds value of element <code>Token</code> from <code>SetExpressCheckout</code> response as a name/value pair to the following URL, and redirects the user's browser to it: <code>https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout&amp;token=<b>value</b></code>	
Logs in to PayPal, approves the use of PayPal, and clicks <b>Continue</b> or <b>Pay</b> .		4. With a GET, redirects user's browser to merchant's <code>ReturnURL</code> with token value appended.
	5. Optionally calls <code>GetExpressCheckoutDetails</code> API with <code>Token</code> to retrieve customer's information.	Returns <code>GetExpressCheckoutDetails</code> response with <code>PayerID</code> , email address, shipping address, confirmed or unconfirmed status of that shipping address, and other details.
	Renders page in customer's browser for the next step in checkout process, such as your "Order Review" page.	
Clicks "Confirm Order" button	6. Calls <code>DoExpressCheckoutPayment</code> API with the required elements <code>Token</code> , <code>OrderTotal</code> , <code>PaymentAction</code> , and <code>PayerID</code> returned by <code>GetExpressCheckoutDetails</code> response.	Returns payment info with important <code>TransactionID</code> value and other details about the payment.

### 3.2 Express checkout as payment mark

In this case, the EC API is used in order to do a standard PayPal payment, using the advantages of the synchronous API process. The PayPal payment happens at the end of the merchant's flow where the user has already registered or provided his shipment information. In this case the call to "**GetExpressCheckoutDetails**" is optional.

For this case, two flows are possible, where the main difference is the perception of the payment by the customer.

### 3.2.1 EC with landing Page :

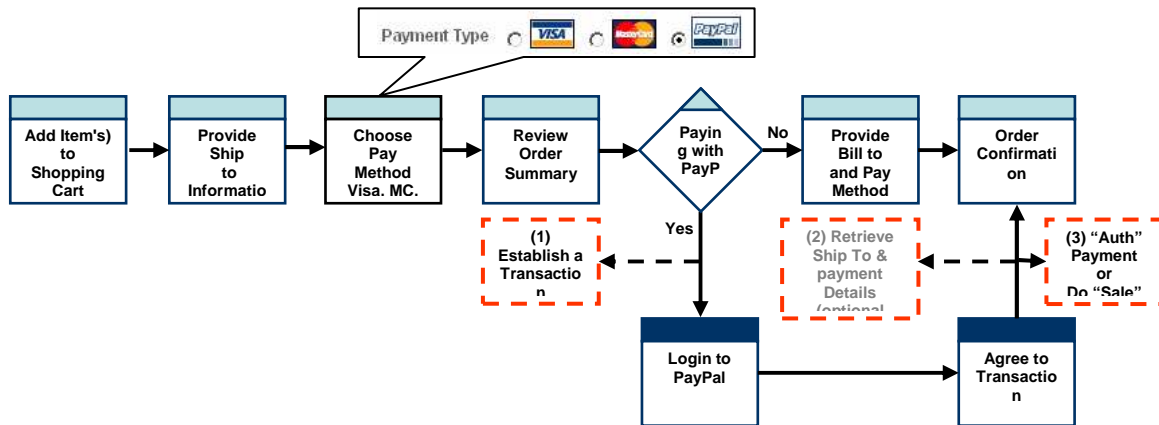


In this case, the customer comes back to a landing page at the merchant. The merchant can display payment information, couponing possibilities, up sell, ...

The payment is only done once the call to **“DoExpressCheckoutPayment”** is done with success.

The call to **“GetExpressCheckoutDetails”** is optional. But it can be used to retrieve pass-through variables that we provide: in the first call, the merchant can use the fields **“InvoiceID”** or **“Custom”** to pass some information that will come back in **“GetExpresCheckoutDetails”**. The difference between **“InvoiceID”** and **“Custom”** is that **“InvoiceID”** is stored by PayPal and shown in the transaction report. These fields can be used for example to pass a session identifier to PayPal in order to rebuild the session once the customer comes back from PayPal.

### 3.2.2 EC without landing Page



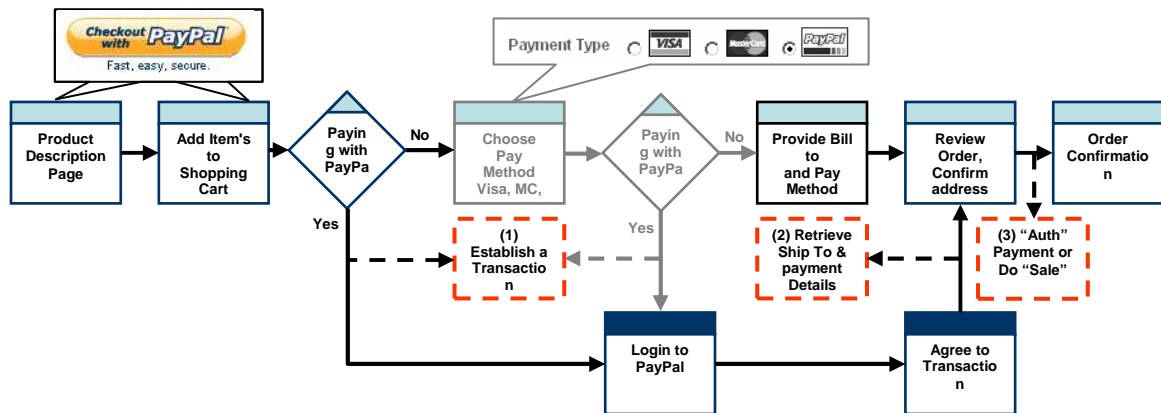
In this case, the review order page is displayed before the user starts the payment process with PayPal. Once the customer confirms the payment, he is redirected to the merchant, which does the call to **“DoExpressCheckoutPayment”** immediately and then displays the order confirmation page.

The call to **“GetExpressCheckoutDetails”** is optional. But it can be used to retrieve pass-through variables that we provide: in the first call, the merchant can use the fields **“InvoiceID”** or **“Custom”** to pass some information that will come back in **“GetExpresCheckoutDetails”**. The difference between **“InvoiceID”** and **“Custom”** is that **“InvoiceID”** is stored by PayPal and shown in the transaction report. These fields can be used for example to pass a session identifier to PayPal in order to rebuild the session once the customer comes back from PayPal.

In this case, it also possible to change the button on the PayPal page. It can be changed from **“Review”** to **“Pay”**. How to do this is described in the **“SetExpressCheckout”** API definition below.

### 3.3 Express checkout as Shortcut

The Express checkout API is used in this case to perform a PayPal payment before the customer creates his account or log into the merchant's website.



The payment button can be placed on the cart page or also directly on the product description page.

During the payment flow, the customer selects on PayPal the address he wants to use to the shipping.

The call to **“GetExpressCheckoutDetails”** allows you to retrieve all the shipping or billing information of the customer. At this stage, the payment is not yet done.

With this information, it is possible to calculated the shipping costs or to offer to the customer to open a account.

Once he confirms the order, the call to **“DoExpressCheckoutPayment”** will do the payment.

## 4 Express Checkout API definition

### 4.1 Security Parameters

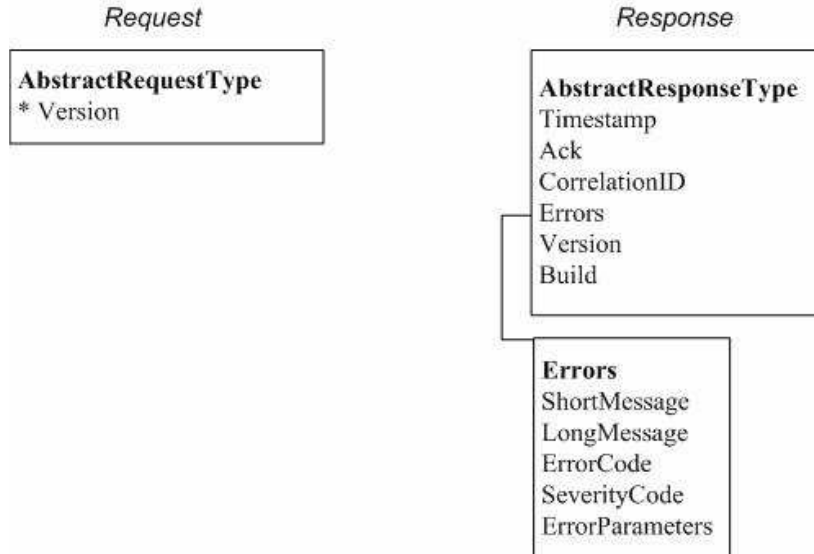
For the security of your business, PayPal must verify that merchants or third-party developers are permitted to initiate a transaction before they make one. PayPal authenticates each request.

The elements are:

Parameter	Optional or Required	Value
Username	Required	Your PayPal API Username.
Password	Required	Your PayPal API Password.
Signature	Optional	Your API signature string. NOTE: If you use an API Certificate, do not include this parameter.
Subject	Optional	Email address of a PayPal account that has granted you permission to make this call. NOTE: Set this parameter only if you are calling an API on a different user's behalf.

## 4.2 Base definition of all API calls

The following types are the base definition of all call and responses.



The response fields are:

- **Timestamp** The date and time (UTC/GMT) the response was generated by the PayPal service as a result of processing a request
- **Ack** A token representing the application-level acknowledgement code. Contains one of the following enumerated values: **Success**, **Failure**, **SuccessWithWarning**, **FailureWithWarning**, **CustomCode**
- **Correlation ID** A string that identifies the exact API request that produced this response. This string is used by PayPal to trace transactions.
- **Errors** Error code and messages for debugging a response :**ErrorCode**, **ShortMessage**, **LongMessage**, **ErrorParameters**, **SeverityCode**
- **Version** A string representing the version of the PayPal API.
- **Build** A string representing the specific software build that processes the request and generates the response.

### 4.3 SetExpressCheckout

The following table details the fields used by this function.

Please refer to “PP\_APIReference.pdf” or “PP\_ExpressCheckout\_IntegrationGuide.pdf” for a detailed description of all fields.

Depending on which API (SOAP or NVP) or SDK you want to use, please refer to the corresponding guides for the definition of the header and generic responses that are handling all the different transaction types.

Request	Response
<div> <b>SetExpressCheckoutRequestDetailsType</b>            * OrderTotal            * ReturnURL            * CancelURL            Token            MaxAmount            OrderDescription            Custom            InvoiceID            ReqConfirmShipping            NoShipping            AddressOverride            LocaleCode            PageStyle            cpp-header-image            cpp-header-border-color            cpp-header-back-color            cpp-payflow-color            Address            PaymentAction            BuyerEmail            SolutionType            LandingPage            ChannelType         </div>	<div> <b>SetExpressCheckoutResponseType</b>            Token         </div>
<div> <b>PaymentActionCodeType</b>            Authorization            Order            Sale         </div>	
<div> <b>AddressType</b>            Name            * Street1            Street2            * CityName            * StateOrProvince            * PostalCode            * Country         </div>	





The required fields are:

- **OrderTotal** : The total cost of the order of the customer. The amount must have two decimal places and the currency needs to be set using the three letter currency code.
- **ReturnURL**: URL to which the customer's browser is returned after choosing to pay with PayPal.
- **CancelURL**: URL to which the customer is returned if he does not approve the use of PayPal to pay you.

The following fields are mandatory.

- **Custom**: A free-form field for your own use, such as a tracking number or other value you want PayPal to return on “**GetExpressCheckoutDetails**” response and “**DoExpressCheckoutPayment**” response.
- **InvoiceID**: Your own unique invoice or tracking number. PayPal returns this value to you on “**DoExpressCheckoutPayment**” response. This value is also shown in the reports
- **NoShipping**: If this field is set to “1”, on the PayPal payment review page, the customer will not see any fields about shipping address. Default value is “0” (shipping displayed).
- **LocaleCode**: Language code of the language that will be used by PayPal to display the first page during Express Checkout. By default, it is the language of the merchant PayPal account.
- **Adress**: If you already know the shipping address of the customer, you may provide it into the address fields in order to either prefill the fields in case the user opens a PayPal account or to display the selected shipping in PayPal.

Please refer to “PP\_APIReference.pdf” or “PP\_ExpressCheckout\_IntegrationGuide.pdf” for a detailed description of all fields.

After you receive a successful response from PayPal, you should add the value of the Token from **SetExpressCheckoutResponse** as a name/value pair to the following URL, and redirect your customer's browser to it:

[https://www.paypal.com/cgi-bin/webscr?cmd=\\_express-checkout&token=value\\_from\\_SetExpressCheckoutResponse](https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout&token=value_from_SetExpressCheckoutResponse)

In order to change the button text of the PayPal page from “review” to “Pay”, you need to add “useraction=commit” at the end of the redirect URL :

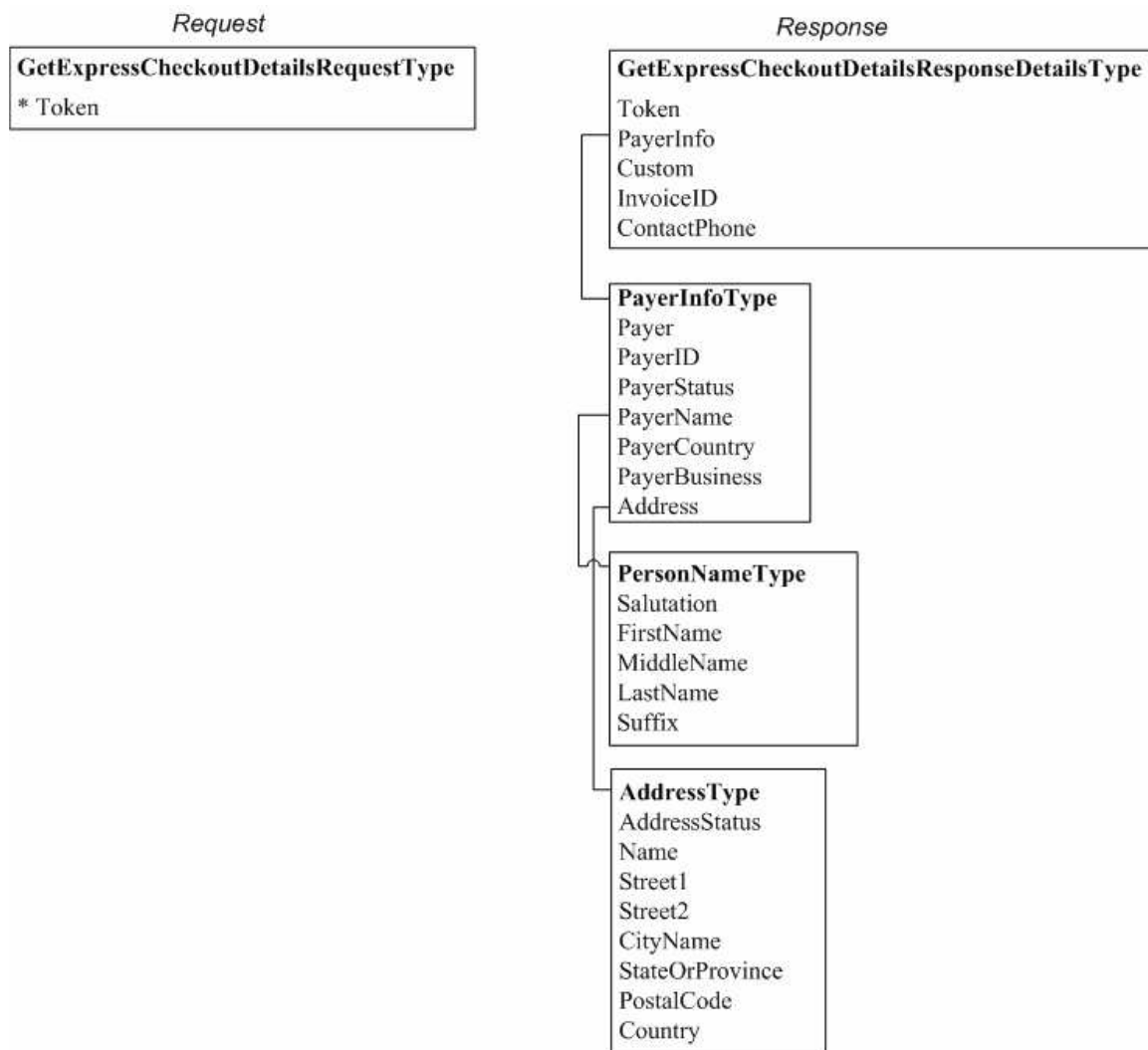
[https://www.paypal.com/cgi-bin/webscr?cmd=\\_express-checkout&token=value\\_from\\_SetExpressCheckoutResponse&useraction=commit](https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout&token=value_from_SetExpressCheckoutResponse&useraction=commit)

## 4.4 GetExpressCheckoutDetails

The following table details the fields used by this function.

Please refer to “PP\_APIReference.pdf” or “PP\_ExpressCheckout\_IntegrationGuide.pdf” for a detailed description of all fields.

Depending on which API (SOAP or NVP) or SDK you want to use, please refer to the corresponding guides for the definition of the header and generic responses that are handling all the different transaction types.



After the customer reviewed the payment at PayPal, the customer is redirected to the URL provided in the “**ReturnURL**” field and happened to it two more field: “**Token**” and “**PayerID**”.



“**GetExpressCheckoutDetails**” uses the value of “**Token**” to retrieve the customer’s information.

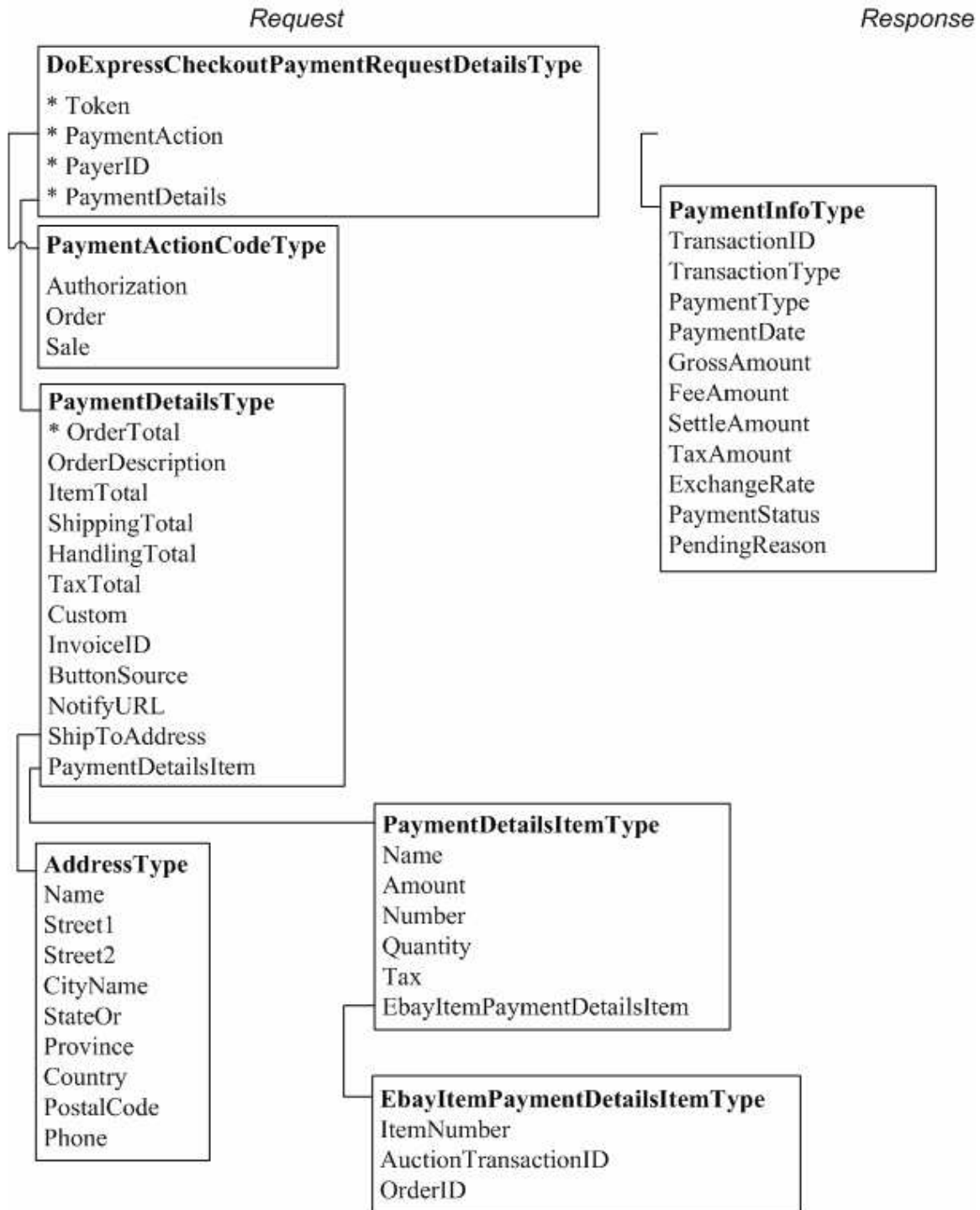
This function can also be used to retrieve the values of the fields “**Custom**” and “**InvoiceID**”.

## ***4.5 DoExpressCheckoutPayment***

The following table details the fields used by this function.

Please refer to “PP\_APIReference.pdf” or “PP\_ExpressCheckout\_IntegrationGuide.pdf” for a detailed description of all fields.

Depending on which API (SOAP or NVP) or SDK you want to use, please refer to the corresponding guides for the definition of the header and generic responses that are handling all the different transaction types.



After the customer reviewed the payment at PayPal, the customer is redirected to the URL provided in the “**ReturnURL**” field and happened to it two more field: “**Token**” and “**PayerID**”.



The required fields are:

- **Token :** This value is returned by PayPal in the call of the return URL.
- **PayerID:** This value is returned by PayPal in the call of the return URL.
- **OrderTotal** Total of order, including shipping, handling, and tax. This is the amount the customer will be charged. The amount must have two decimal places and the currency needs to be set using the three letter currency code.
- **PaymentAction:** How you want to obtain payment:
  - **Authorization** indicates that this payment is a basic authorization subject to settlement with PayPal Authorization & Capture.
  - **Order** indicates that this payment is is an order authorization subject to settlement with PayPal Authorization & Capture.
  - **Sale** indicates that this is a final sale for which you are requesting payment. The funds are immediately moved from the customer PayPal account to the merchant PayPal account

Please refer to “PP\_APIReference.pdf” or “PP\_ExpressCheckout\_IntegrationGuide.pdf” for a detailed description of all fields.



## 5 Useful links

**Developer central/sandbox**

<https://developer.paypal.com>

**Soap API and SDK :**

[https://www.paypal.com/IntegrationCenter/ic\\_sdk-resource.html](https://www.paypal.com/IntegrationCenter/ic_sdk-resource.html)

**NVP API and SDK :**

[https://www.paypal.com/IntegrationCenter/ic\\_nvp.html](https://www.paypal.com/IntegrationCenter/ic_nvp.html)

**PayPals developer forum :**

<http://www.pdncommunity.com/>