# Payment Gateway

## Table of Contents

[Payment Gateway 1](#_Toc42368393)

[1. Introduction 4](#_Toc42368394)

[2. What is Payment Gateway [1] 5](#_Toc42368395)

[3. What is the Role of a Payment Gateway 5](#_Toc42368396)

[4. How Does Payment Gateway Work 5](#_Toc42368397)

[5. Payment Gateway Architecture 6](#_Toc42368398)

[6. What to look for in a Payment Gateway? 6](#_Toc42368399)

[7. Security for merchant and security for customer: 7](#_Toc42368400)

[8. Benefits of payment gateway 7](#_Toc42368401)

[9. What is Payment Gateway Charges 8](#_Toc42368402)

[10. Examples payment gateways 9](#_Toc42368403)

[11. Create a Payment Gateways (High-level Architecture) [2] 10](#_Toc42368404)

[a) Security 11](#_Toc42368405)

[b) Simplified Controller 11](#_Toc42368406)

[c) Testing 13](#_Toc42368407)

[d) Potential caveats 14](#_Toc42368408)

[12. Conclusion 15](#_Toc42368409)

[References 16](#_Toc42368410)

## Introduction

E-commerce payment gateway is the access point to the online banking network. All online transactions must pass through a payment gateway to be processed. The payment gateways act as a bridge between the user’s website and the financial institutions that process the transaction. Gateways process the different transactions between user and web browser. A payment gateway authenticates and routes payment. In this paper, we will show the different e-commerce payment gateways, and an example of how to implement Payment Gateways. An Internet e-commerce payment gateway is a critical infrastructural component to ensure that such transactions occur without any hitches and in total security over electronic networks.

Payment gateway/Payment system/Payment method is the three interchanging words; however, payments gateway differs from the two in terms of definition. A payment gateway creates a connection between Payer and Payee over the internet. Payer knew as payment sender while Payee knew as payment receiver, in other words, the payee is the merchant however payer is the consumer

## What is Payment Gateway [1]

A payment gateway is an online payments’ service that, when integrated with the e-commerce platform, is devised as the channel to make and receive payments.

The procedure to receive payments includes the customer requiring to fill in some details, like credit/debit card number, expiry date, and CVV. Post this, the customer proceeds to make a payment, which then, gets transferred from the buyer’s account to the seller’s (merchant’s) account.

## What is the Role of a Payment Gateway

The main role of an online Payment Gateway is to approve the transaction process between merchant and customer. It plays a vital role in the online transaction process and authorizes transactions between merchants and customers.

It helps the e-commerce platform aggravate its existence with ease of payments to offer to its customers. Besides, it also leads to the e-commerce platform gaining rapport for leading to not only quick and secure payments but also convenience and success with the same every time.

A payment gateway service can be provided by banks directly or a payment service provider authorized by a bank.

## How Does Payment Gateway Work

Straight away coming to the functioning of an online payment gateway, it follows a procedure for settling the payment every time. This happens when a customer places the order for a service/product from a payment gateway-enabled merchant. From filling in the card details to payment finally flowing into the merchant’s account and settling, the payment gateway passes through a variety of steps-

* **STEP 1:** After the customer places the order online and proceeds to make payment for the same, he/she needs to enter credit/debit card details.
* **STEP 2:** The card details are encrypted in a secure way with Secure Socket Layer (SSL) encryption to be sent between the browser and the merchant’s web server. A payment gateway eliminates the merchant’s Payment Card Industry Data Security Standard (PCI DSS) compliance obligations without redirecting customers away from the website.
* **STEP 3:** After this, the merchant forwards transaction details to their payment gateway, which is also an SSL encrypted connection to the payment server hosted by the payment gateway.
* **STEP 4:** The payment gateway converts the message from XML to ISO 8583 or a variant message format (format understood by EFT Switches) and then forwards the transaction information to the payment processor used by the merchant’s acquiring bank.
* **STEP 5:** The payment processor forwards the transaction information to the card association (I.e.: Visa/MasterCard/American Express).
* **STEP 6:** Next, the credit card issuing bank receives the authorization request, verifies the credit or debit available and then sends a response back to the processor (via the process same as for the authorization) with a response code (i.e., approved or denied). The response code also helps to communicate the reason for the case of a failed transaction, for example, insufficient funds, and so on.
* **STEP 7:** The processor then forwards the authorization response to the payment gateway, and the payment gateway receives the response and forwards it onto the interface used to process the payment. This process is termed as Authorization or “Auth”. This entirely takes around 2-3 seconds in general.
* **STEP 8:** The merchant then fulfills the order and the above process can be repeated but this time to “Clear” the authorization by consummating the transaction. Typically, the “Clear” is initiated only after the merchant has fulfilled the transaction (I.e. shipped the order). This results in the issuing bank ‘clearing’ the ‘auth’ (I.e. moves auth-hold to a debit) and prepares them to settle with the merchant acquiring bank.
* **STEP 9:** The merchant submits all their approved authorizations, in a “batch” (end of the day), to their acquiring bank for settlement via its processor. This typically reduces or “Clears” the corresponding “Auth” if it has not been explicitly “Cleared.”
* **STEP 10:** The acquiring bank makes the batch settlement request of the credit card issuer.
* **STEP 11:** The credit card issuer makes a settlement payment to the acquiring bank (the next day in most cases).
* **STEP 12:** The acquiring bank subsequently deposits the total of the approved funds in to the merchant’s nominated account (the same day or next day). This could be an account with the acquiring bank if the merchant does their banking with the same bank or an account with another bank.

## Payment Gateway Architecture

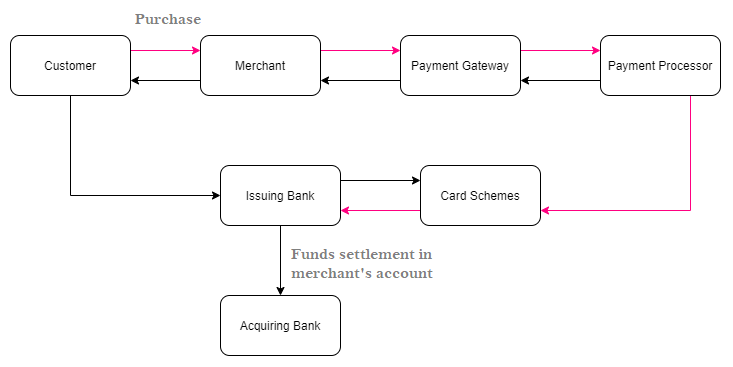


Figure 1: Payment Gateway Architecture

## What to look for in a Payment Gateway?

How To Pick The Right Payment Gateway For Your Ecommerce Website

Any business that collects online payment or accepts credit/ debit card payments, needs an online Payment Gateway. It’s important to research and find the correct payment gateway provider that fits your unique needs. A payment gateway should offer numerous benefits, namely, quick settlement of payments, the hassle-free flow of transactions, and a superb overall experience every time. So, what more is there to look for in a Payment Gateway?

* **Security:** A secure transaction is a first and foremost need for any business that collects payments online. Customers must trust in the transaction and that their personal and financial information will be safe while online transacting. It should ensure the security of the information a customer put in. Such a service should be compliant with PCI standards.
* **Customer experience:** A payment gateway should provide unique and swift payment processing and payment customization. It should be mobile optimized, secure and convenient which will provide customers with a seamless shopping experience.
* **Worldwide Payment Acceptance:** We live and work in a global market. A payment gateway should support multi-currency transactions.
* **Merchant Onboarding:** A payment gateway should provide the quickest merchant onboarding, as online transactions are a primary need in today’s world. The processing and setup fees and rates should fit in the merchant’s budget.
* **Payment Options:** Last but definitely not the least, a payment gateway should offer multiple payment options. A customer should be able to choose a payment method he/she is most comfortable with, be it Credit/Debit Cards, Netbanking, Mobile Wallets, UPI or any other.

Here are some ways with which a payment gateway keeps information secure

## Security for merchant and security for customer:

A payment gateway ensures the security of the information you put in. Here is a list of things that an online Payment Gateway does to keep your data safe:

The standard security protocol used in online transactions is SSL(Standard Security Layer) It protects sensitive card information and authenticates the customer’s identity. A payment gateway with SSL can be identified by checking the ‘https’ at the beginning of the web address.

To authenticate customers and merchants an additional layer of security can be implemented under the 3ds(Three-Domain Secure) protocol. This is a messaging protocol developed by EMVCo.

Data encryption is one of the most important security measures in the payment gateway, where the data appears scrambled and illegible to anyone but you.

Another unique way for a secure transaction via payment gateway is tokenization, where sensitive card details are replaced by a string of encrypted characters.

## Benefits of payment gateway

Gone are the days when, a Payment Gateway was used to be a tool for transferring money, now a Payment Gateway can do a lot more than that, want to know? Here is a glimpse:

* White label
* Fraud and risk management
* PCI DSS Wallet
* API tools for easy integration

## What is Payment Gateway Charges

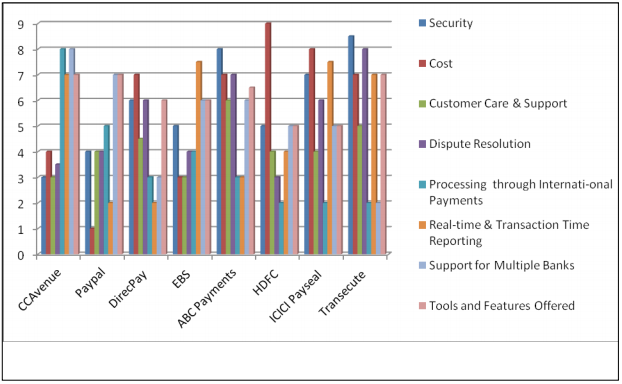
Charges for payment gateway generally depend on many elements like,

* Business model
* Features and services merchants require from a payment gateway
* Set up fee
* Software fee
* Maintenance fee
* Transaction discount rate according to the payment mode

With zero setup fee and quickest merchant onboarding, opt for Lyra’s Payment Gateway. Accept payments online for your business globally with 100+ payment options. Keep your digital transactions safe and your customers happy

## Examples payment gateways

This below graph shows the various factors of different payment gateways. This graph differentiate here different payment gateways with various factors through the different colors. So according to this analysis we can easily understand that which is best option for the user.

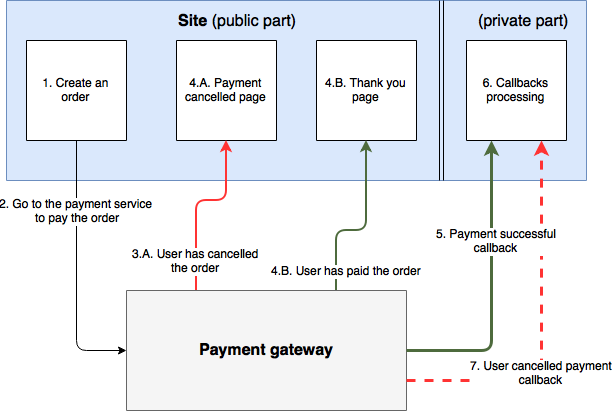


## Create a Payment Gateways(High-level Architecture) [2]



At some point we all face the need to integrate a payment gateway. It may seem difficult, but actually it's not. There are many payment gateways to choose from, but the good point is that most of them work the same way. You just need to understand the high-level picture and you're ready to integrate any of them into your project.

As I mentioned before, there are many payment systems, but basically they all work in the same way:



1. A user visits the site and creates an order. It may be anything: goods, services and so on. After this step, you have a placed order in your own system.

2. The user is redirected to the payment gateway page where he can choose a payment method and pay for the order.

3.A. If the user cancels the payment, he will be redirected to the cancel URL.

4.A. It may be any page of your site, for example, a home page or a page where you ask the user about any issues with payment and motivate him to complete the order.

3.B. If the user pays, then he will be redirected to the success URL.

4.B. At this page, I usually tell the user a few warm words like "Thank you". Keep in mind: when a user lands on this page, it doesn't mean that you got the real payment. So, you may write something like: "Thank you for choosing us. The service will be provided as soon as the payment processing is done". Never rely on this page to decide if you should provide a service to the user right away.

5. After the real payment is done, the payment gateway will notify your callback URL.

6. This page isn't for public use. On this page, you must process callbacks from the payment gateway. They send you all necessary data to understand what order has been paid. For example, they send you the order id in your own system, transaction id in their system, amount, currency and so on.

7. Some payment systems allow users to cancel the payment. This means that you may get such a callback and your site must handle it properly. For example, reject the service for the user, update payment statistics in your system and so on.

### Security

1. Your callback URL must be accessible to non authenticated users. In Symfony we can achieve this via:

security:

1. security:
2. firewalls:
3. payment\_callbacks:
4. pattern: ^/paymentcallback/
5. security: false

2. Some payment systems provide lists of white IPs that you must check for to treat a request as a legal one.

3. Almost all payment gateways encode data they transfer with private keys.

### Simplified Controller

Again, I'm going to use Symfony for all examples, but they are simple and you can easily understand them and modify the code sample for your own framework and language. All actions are described below the code sample.

1. <?php
3. // ...
5. class OrderController extends Controller
6. {
7. /\*\*
8. \* @Route("/pay/{hash}", name="OrderView")
9. \* @Method({"GET"})
10. \*/
11. public function payOrderAction($hash)
12. {
13. $order = // get order by hash or ID
14. $payUrl = // This URL you will form it using parameters like order id in your system, amount, currency and so on. It depends on the particular payment system you work with. Please check their docs.
16. return new RedirectResponse($url);
17. }
19. /\*\*
20. \* @Route("/thank-you/{orderId}", name="OrderThankYou", requirements={"orderId": "\d+"})
21. \* @Method({"GET"})
22. \* @Template()
23. \*/
24. public function thankYouAction($orderId)
25. {
26. return [
27. 'id' => $orderId,
28. ];
29. }
31. /\*\*
32. \* @Route("/paymentcallback/custom-super-secret-endpoint", name="PaymentCallback")
33. \*/
34. public function callbackAction(Request $request)
35. {
36. // log callback
37. // get data from $request ($\_GET, $\_POST...)
38. $isValid = // validate data from the request
39. if ($isValid) {
40. // give the user his service
41. }
43. // return a response to the payment gateway. Usually it's a simple 'OK', 200 status code.
44. return new Response('OK');
45. }
46. }

So, we have three main actions:

**1. Pay Order.**

Here we form the URL for the user. At this URL, the user will see the payment method or methods to choose from. Usually such URL must contain parameters like:

orderId - order ID in your system

amount - payment amount

currency - currency, like USD, EUR and so on.

email/userId - a user email or id in your system

accepturl - we set up here a URL to the Thank You page. I also put there the order id to be able to show more information to the user.

cancelurl - just points to the home page or any other page you want

callbackurl - URL for payment gateway callbacks

**2. Thank You**

This page is to notify the user that he will get his service as soon as the payment processing is done.

**3. Callback**

Only here you process the result of a real payment. First of all, you need to get request data, then validate it and if everything is OK, then you have to provide the user his service and return the 200 response. Usually a response to a callback must containt simple work like "OK" or something similar. You will find this in the payment gateway documentation. You can also dispatch an event and attach any useful listeners to it. For example, you may send an email to the user, update payment stats and so on.

### Testing

When you start, you develop new things on your local machine. You need to be sure that everything works fine before you go live and that's why you need ability to test payments locally. Many of payment systems allow to make test payments without using real money. Usually you need to complete a few steps to enable testing in a payment gateway:

1. Enable the test mode on their side via UI.

2. Allow requests from any sites to the payment gateway endpoint. Many payment systems accept requests only from a set of configured in your account domains.

3. Finally, you need to provide a special parameter in the URL, like "test".

Ok, now you know what to do to enable test mode, but we still have one problem. As I mentioned before, after you complete payment, a payment gateway will ping your server (remember callback URL). As you make payments from a local machine, they won't be able to reach your local host.

Here payment gateways (not all of them) provide some useful options:

1. In their admin panel they have a simple HTML form and using it you can imitate any kind of a callback request. This is one of the best options, because you won't have any issues with your local domain (you just submit a form via browser and your local domain will work fine). You don't have to do any additional programming.

2. Some of them add all data they would return at callback to the success/accept URL. This is useful, too. You just can modify your "Thank you" action a little:

That's it. This way you will be able to test payments locally in a test mode.

### Potential caveats

* documentation is outdated. Yes, this still happens. In such cases you need to contact their tech guys or think twice if you still want to integrate such payment gateway.
* zero instruments to test payments before go live. If this is your case, you can log their callbacks to your production server or make a payment gateway available only to a test user.
* personal data approvement before you can use a gateway even in a test mode
* a gateway doesn’t support recurring payments

Conclusions

Don't be afraid, integration is easy no matter what gateway you plan to choose.

## Conclusion

Through above comparative study of different gateways user can access it easily and he can take decision that which e-commerce payment is better for him because people wants a user friendly payment gateway which should be easy to use and easy to access. These payment systems with comparative study are useful for users because some people don’t know about payment gateways so they can get knowledge about it and after using gateways they feel trustee it. It is also necessary in payment system that assures privacy in the payments like bank accounts and there should be a way of identifying the customer’s bank accounts and the merchant bank accounts.

## References

|  |  |
| --- | --- |
| [1] | l. teams, "What is Payment Gateway and How does it Work," 16 May 2019. [Online]. Available: https://lyra.com/in/what-is-payment-gateway/. |
| [2] | victor, "High-level Architecture of Payment Gateways," [Online]. Available: https://victor.4devs.io/en/payment-gateways/high-level-architecture-of-payment-gateways.html. |