

Merchant Gateways



Download the SDK

The [SDK](#) includes a sample Merchant Gateway to get you started.

Merchant Gateways must extend the `MerchantGateway` class and implement at least one of the four interfaces provided by the payment system, to ensure proper functionality and integration.

MerchantGateway

The **MerchantGateway** class is an abstract class that defines methods that all merchant gateways must implement. In addition, it implements a set of overwritable helper methods to handle common errors as well as HTTP requests, and extends the **Gateway** class which can be found in `/installpath/components/gateways/lib/gateway.php`.

The `MerchantGateway` class can be found in `/installpath/components/gateways/lib/merchant_gateway.php`.

Common Errors

When implementing merchant gateways a number of errors may occur either with validation or with processing. The `MerchantGateway` class provides a method called `getCommonError()`, which converts common error types into error response values suitable for error message passing (i.e. can be set into `Input::setErrors()`). You are encouraged, whenever possible, to use these common error types.

Type	Usage
<code>card_number_invalid</code>	Use whenever a credit card number is invalid.
<code>card_expired</code>	Use whenever a credit card has expired.
<code>routing_number_invalid</code>	Use whenever a routing number is invalid.
<code>account_number_invalid</code>	Use whenever an account number is invalid.
<code>duplicate_transaction</code>	Use whenever a duplicate transaction has been detected.
<code>card_not_accepted</code>	Use whenever a credit card type is not accepted.
<code>invalid_security_code</code>	Use whenever the credit card security code provided is invalid.
<code>address_verification_failed</code>	Use whenever the address provided is invalid.
<code>transaction_not_found</code>	Use whenever processing a capture, refund, or void and the original transaction can not be found.
<code>unsupported</code>	Use whenever the gateway does not support the requested action (e.g. <code>refundCc()</code>)
<code>general</code>	Use for all other cases.



Declined responses are not errors

Do not trigger an error response for declined payments, Blesta will automatically issue the appropriate error.

Setting Errors

```
$this->Input->setErrors($this->getCommonError("unsupported"));
```

Merchant Interfaces

There are four merchant gateway interfaces, each with its own unique set of method requirements. A payment gateway must implement at least one of these interfaces.

- **MerchantCc** found in `/installpath/components/gateways/lib/merchant_cc.php`
- **MerchantAch** found in `/installpath/components/gateways/lib/merchant_ach.php`
- **MerchantCcOffsite** found in `/installpath/components/gateways/lib/merchant_cc_offsite.php`
- **MerchantAchOffsite** found in `/installpath/components/gateways/lib/merchant_ach_offsite.php`
- **MerchantCcForm** found in `/installpath/components/gateways/lib/merchant_cc_form.php`
- **MerchantAchForm** found in `/installpath/components/gateways/lib/merchant_ach_form.php`

Both the **MerchantCc** and **MerchantAch** interfaces allow for the capturing, returning, and voiding of funds with account information stored within Blesta. The MerchantCc interface handles credit cards, while the MerchantAch interface handles bank account transfers.

The **MerchantCcOffsite** and **MerchantAchOffsite** interfaces allow for the capturing, returning, and voiding of funds stored with the gateway processor. In addition, these interfaces define methods allowing payment account details to be stored, updated, and removed from the gateway processor. Again, the MerchantCcOffsite interface handles credit cards, while the MerchantAchOffsite interfaces handles bank account transfers.

The **MerchantCcForm** and **MerchantAchForm** interface allows for the replacement of credit card or ach forms in Blesta with one generated by the gateway. This is typically for use in authorizing payments using tokens for credit cards or bank accounts that are not sent through Blesta.

The following table will help identify which interfaces should be implemented for a payment gateway based upon the features of that payment processor.

Processor Feature	MerchantCc	MerchantAch	MerchantCcOffsite	MerchantAchOffsite	MerchantCcForm	MerchantAchForm
Accept Credit Cards	Yes	No	Maybe	No	Yes	No
Accept Bank Transfers	No	Yes	No	Maybe	No	Yes
Allow Credit Card Account Storage	Maybe	No	Yes	No	Maybe	No
Allow Bank Account Storage	No	Maybe	No	Yes	No	Maybe
Allow Tokenized Credit Cards	Maybe	No	Maybe	No	Yes	No
Allow Client-Side Only Forms	No	No	No	No	Yes	Yes