

GET STARTED

Overview

Set Up Your Client

Simple Server →

Go Live

CHECKOUT UIS

Drop-in UI

Hosted Fields

TUTORIALS & EXAMPLES

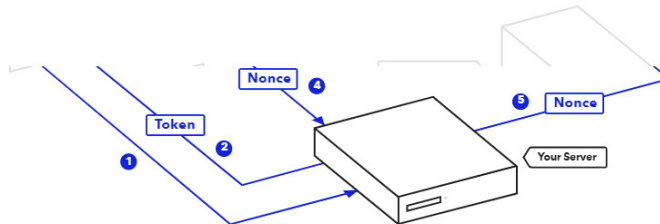
Drop-in Tutorial

Hosted Fields Tutorial

Example Integrations with
Drop-in

Set Up Your Server

PHP SDK ▾



Install and configure

AVAILABILITY

The Braintree PHP SDK requires PHP version 7.2.0 or higher and the PHP cURL extension.

Get started with our PHP library by hitting the download link below.

[Download PHP library](#)

- Version: 4.5.0
- SHA256: `db1a7572944a221b2d661aaa1f9276d2a9b44ccb306589456ec8a6ce489f2e99`

Or use composer

[Composer](#) is a package manager for PHP. In the `composer.json` file in your project add:

```
JavaScript Copy
```

```
{
  "require": {
    "braintree/braintree_php": "4.5.0"
  }
}
```

And then run:

```
php composer.phar install
```

In your code, configure the environment and [API credentials](#):

```
PHP Copy
```


```
$gateway = new Braintree_Gateway([
  'environment' => 'sandbox',
  'merchantId' => 'use_your_merchant_id',
  'publicKey' => 'use_your_public_key',
  'privateKey' => 'use_your_private_key'
]);
```

⚠ This code snippet now uses gateway instance methods instead of class-level methods. [Learn more.](#)


NOTESee the [Braintree PHP SDK Version Changelog](#).

Generate a client token

Your server is responsible for [generating a client token](#), which contains all authorization and configuration information your client needs to initialize the client SDK to communicate with Braintree. Including a [customerId](#) when generating the client token lets returning customers select from previously used payment method options, improving user experience over multiple checkouts.

```
PHP Copy 
```

```
$clientToken = $gateway->clientToken()->generate([
    "customerId" => $aCustomerId
]);
```


 This code snippet now uses gateway instance methods instead of class-level methods. [Learn more.](#)

If the customer can't be found, it will return a [validation error](#).


[Set Up Your Client](#) covers the client side of the exchange.

Send a client token to your client

Here is an example of how your server would generate and expose a client token:

```
PHP Copy 
```

```
echo($clientToken = $gateway->clientToken()->generate());
```


 This code snippet now uses gateway instance methods instead of class-level methods. [Learn more.](#)

How the token is used by the client may vary. In JavaScript integrations the client token is often included in the generated HTML/JS, while in mobile apps the client token must be requested. These methods are discussed in the [client token setup section](#).

Receive a payment method nonce from your client

Once your client successfully obtains a customer payment method, it receives a `payment_method_nonce` representing customer payment authorization, which it then sends to your server.


Your server implementation is then responsible for receiving the `payment_method_nonce` and using it appropriately.

```
PHP Copy 
```


```
$nonceFromTheClient = $_POST["payment_method_nonce"]
/* Use payment method nonce here */
```

Create a transaction

You can create a transaction using an `$amount` and the `$nonceFromTheClient` you received in the previous step:

```
PHP Copy 
```

```
$result = $gateway->transaction()->sale([
    'amount' => '10.00',
    'paymentMethodNonce' => $nonceFromTheClient,
    'options' => [
        'submitForSettlement' => True
    ]
]);
```

 This code snippet now uses gateway instance methods instead of class-level methods. [Learn more.](#)

The sale call returns a [Transaction Result Object](#) which contains the transaction and information about the request.

Test your integration

See our [Testing page](#) for values you can use for `$nonceFromTheClient` in your sandbox account. These nonces can be passed as strings through server-side calls to generate payment methods in the desired state. To verify your integration, you can check in the [sandbox Control Panel](#), where transactions will immediately appear on success.

[sandbox account](#), where transactions will immediately appear on success.

IMPORTANT

Always develop and test your code against your sandbox account before processing live transactions against a production account.

Transition to production

At this point, you should be able to accept a payment method nonce and create a transaction in our sandbox. When you're ready to start charging real money, transition over to our production environment. We'll explain that process next.

[Next Page: Go Live →](#)

Get help from a human

Submit a request for help with your Braintree sandbox or production account.

Get Help

Give us feedback on this page

Submit this short survey to let us know how we can improve your experience.

Send Feedback



Get Started

[Integration Guide](#)
[Tutorial \(Preview\)](#)
[Checkout UIs](#)
[Example Integrations](#)



Basics

[Client Authorization](#)
[Payment Method Nonces](#)
[Transactions](#)
[Customers](#)
[Payment Methods](#)



Payment Method Types

[Overview](#)
[ACH Direct Debit](#)
[Apple Pay](#)
[Credit Cards](#)
[Masterpass](#)
[Google Pay](#)
[PayPal](#)
[Samsung Pay](#)
[UnionPay \(Beta\)](#)
[Venmo](#)
[Visa Checkout](#)



Tools

[3D Secure](#)
[Advanced Fraud Tools](#)
[Client SDK](#)
[Disputes](#)
[Payment Request API](#)
[Reports](#)
[Webhooks](#)



Checkout UI

[Drop-in UI](#)
[Hosted Fields](#)



Additional Features

[Braintree Auth \(Beta\)](#)
[Braintree Marketplace](#)
[Grant API \(Beta\)](#)
[OAuth \(Beta\)](#)
[PayPal Here](#)
[Recurring Billing](#)



API Reference

[Client References](#)
[Server-side API Requests](#)
[Server-side Response Objects](#)
[General](#)