

Stripe Checkout Pages Using PHP

Using Stripe JS and AJAX

Bonus: Simple HTML `<select>` Dropdown

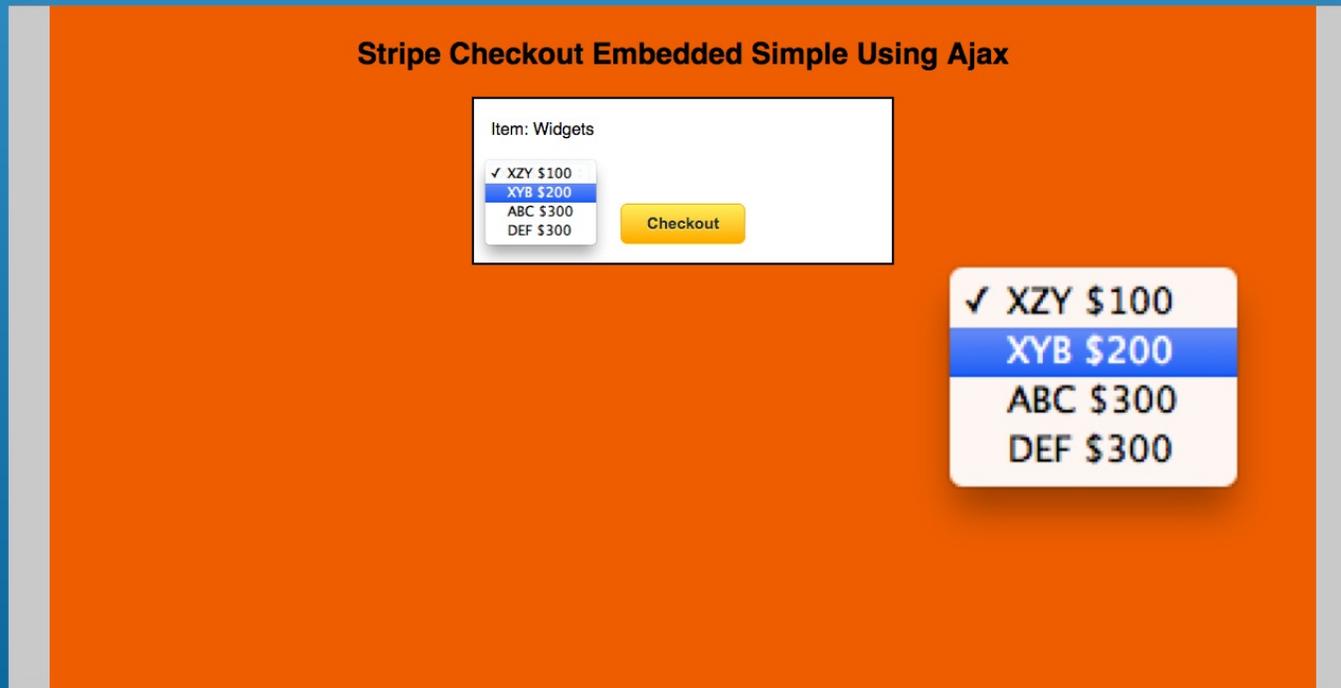
With Lon Hosford

© 2015 Alonzo Hosford

Copyright 2015 Alonzo L. Hosford. All Rights Reserved. www.lonhosford.com

This is a Visual Step by Step Workbook and voice transcript for accompanying video for this portion of the course.

Bonus: Simple HTML `<select>` Dropdown



This is a simplified example of modifying this section's example to allow multiple selections from HTML select element and create a single item checkout. The idea all the data can be statically coded and avoid more complicated programming to store detailed product information or develop a shopping cart.

Statement Descriptor Simplified

BEFORE

```
1 <?php
2 include_once 'common.inc.php';
3 // General website data
4 $company_name = "Acme Widgets Inc.";
5 $bank_statement_descriptor = "ACME-WIDGETS";
6 // Order Data
7 $description = "Widgets";
8 $quantity = 12;
9 $statement_descriptor = $bank_statement_descriptor . ' ' . $quantity . ' ' . $description;
10 $amount = 20.00;
11 ?>
```

checkout_simple.php

AFTER

```
1 <?php
2 include_once 'common.inc.php';
3 // General website data
4 $company_name = "Acme Widgets Inc.";
5 $bank_statement_descriptor = "ACME-WIDGETS";
6 // Order Data
7 $description = "Widgets";
8 $statement_descriptor = $bank_statement_descriptor ;
9 ?>
```

First we simplified the statement descriptor. This is to avoid having to design a statement descriptor to fit into the 22 character limit for every product.

Removed Amount

BEFORE

```
1 <?php
2 include_once 'common.inc.php';
3 // General website data
4 $company_name = "Acme Widgets Inc.";
5 $bank_statement_descriptor = "ACME-WIDGETS";
6 // Order Data
7 $description = "Widgets";
8 $quantity = 12;
9 $statement_descriptor = $bank_statement_descriptor . ' ' . $quantity . ' ' . $description;
10 $amount = 20.00;
11 ?>
```

checkout_simple.php

AFTER

```
1 <?php
2 include_once 'common.inc.php';
3 // General website data
4 $company_name = "Acme Widgets Inc.";
5 $bank_statement_descriptor = "ACME-WIDGETS";
6 // Order Data
7 $description = "Widgets";
8 $statement_descriptor = $bank_statement_descriptor ;
9 ?>
```

Next the amount is removed and we will see later where this is going to be stored.

Removed Quantity

BEFORE

```
1 <?php
2 include_once 'common.inc.php';
3 // General website data
4 $company_name = "Acme Widgets Inc.";
5 $bank_statement_descriptor = "ACME-WIDGETS";
6 // Order Data
7 $description = "Widgets";
8 $quantity = 12;
9 $statement_descriptor = $bank_statement_descriptor . ' ' . $quantity . ' ' . $description;
10 $amount = 20.00;
11 ?>
```

checkout_simple.php

AFTER

```
1 <?php
2 include_once 'common.inc.php';
3 // General website data
4 $company_name = "Acme Widgets Inc.";
5 $bank_statement_descriptor = "ACME-WIDGETS";
6 // Order Data
7 $description = "Widgets";
8 $statement_descriptor = $bank_statement_descriptor ;
9 ?>
```

Arbitrary decision to remove the quantity to simplify the example. So there no quantity choices for the user in this example.

Added select Element

BEFORE

```
24 <div class="form-container center-container">
25   <p>Item: <?=$description?></p>
26   <p>Quantity: <?=$quantity?></p>
27   <input type="hidden" id="stripe-pk" value="<?=$st_test_public_key?>" />
28   <input type="hidden" id="company-name" value="<?=$company_name?>" />
29   <input type="hidden" id="quantity" value="<?=$quantity?>" />
30   <input type="hidden" id="amount" value="<?=$amount?>" />
31   <input type="hidden" id="description" value="<?=$description?>" />
32   <input type="hidden" id="statement-descriptor" value="<?=$statement_descriptor?>" />
```

checkout_simple.php

AFTER

```
22 <div class="form-container center-container">
23   <p>Item: <?=$description?></p>
24   <select id="products">
25     <option value="100">XZY $100</option>
26     <option value="200">XYB $200</option>
27     <option value="300">ABC $300</option>
28     <option value="300">DEF $300</option>
29   </select>
30
31   <input type="hidden" id="stripe-pk" value="<?=$st_test_public_key?>" />
32   <input type="hidden" id="company-name" value="<?=$company_name?>" />
33
34   <input type="hidden" id="amount" value="" />
35   <input type="hidden" id="description" value="" />
36   <input type="hidden" id="statement-descriptor" value="<?=$statement_descriptor?>" />
```

Added a select HTML element using default styling. The select element has an id we can reference in Javascript using JQuery. The option element values are the price. Then the option element text is what the user sees.

Removed Quantity

BEFORE

```
24 <div class="form-container center-container">
25   <p>Item: <?=$description?></p>
26   <p>Quantity: <?=$quantity?></p>
27   <input type="hidden" id="stripe-pk" value="<?=$st_test_public_key?>" />
28   <input type="hidden" id="company-name" value="<?=$company_name?>" />
29   <input type="hidden" id="quantity" value="<?=$quantity?>" />
30   <input type="hidden" id="amount" value="<?=$amount?>" />
31   <input type="hidden" id="description" value="<?=$description?>" />
32   <input type="hidden" id="statement-descriptor" value="<?=$statement_descriptor?>" />
```

checkout_simple.php

AFTER

```
22 <div class="form-container center-container">
23   <p>Item: <?=$description?></p>
24   <select id="products">
25     <option value="100">XZY $100</option>
26     <option value="200">XYB $200</option>
27     <option value="300">ABC $300</option>
28     <option value="300">DEF $300</option>
29   </select>
30
31   <input type="hidden" id="stripe-pk" value="<?=$st_test_public_key?>" />
32   <input type="hidden" id="company-name" value="<?=$company_name?>" />
33
34   <input type="hidden" id="amount" value="" />
35   <input type="hidden" id="description" value="" />
36   <input type="hidden" id="statement-descriptor" value="<?=$statement_descriptor?>" />
```

We no longer need the quantity hidden element.

Purpose of amount and description Hidden Elements

BEFORE

```
24 <div class="form-container center-container">
25   <p>Item: <?=$description?></p>
26   <p>Quantity: <?=$quantity?></p>
27   <input type="hidden" id="stripe-pk" value="<?=$st_test_public_key?>" />
28   <input type="hidden" id="company-name" value="<?=$company_name?>" />
29   <input type="hidden" id="quantity" value="<?=$quantity?>" />
30   <input type="hidden" id="amount" value="<?=$amount?>" />
31   <input type="hidden" id="description" value="<?=$description?>" />
32   <input type="hidden" id="statement-descriptor" value="<?=$statement_descriptor?>" />
```

checkout_simple.php

AFTER

```
22 <div class="form-container center-container">
23   <p>Item: <?=$description?></p>
24   <select id="products">
25     <option value="100">XZY $100</option>
26     <option value="200">XYB $200</option>
27     <option value="300">ABC $300</option>
28     <option value="300">DEF $300</option>
29   </select>
30
31   <input type="hidden" id="stripe-pk" value="<?=$st_test_public_key?>" />
32   <input type="hidden" id="company-name" value="<?=$company_name?>" />
33
34   <input type="hidden" id="amount" value="" />
35   <input type="hidden" id="description" value="" />
36   <input type="hidden" id="statement-descriptor" value="<?=$statement_descriptor?>" />
```

We will populate the amount and description hidden elements using Javascript and JQuery.

Getting the select Element values using JQuery

BEFORE

```
59  /**
60  *   Click event for checkout
61  */
62  $('#checkout-btn').click(function(e){
63      console.log("#checkout-btn");
64      // Open Stripe checkout form
65      stripeFormHandler.open({
66          amount:$('#amount').val() * 100
67      });
68      e.preventDefault();
69  });
```

checkout_ui.js

AFTER

```
58  /**
59  *   Click event for checkout
60  */
61  $('#checkout-btn').click(function(e){
62      console.log("#checkout-btn");
63      console.log($("#products option:selected").val());
64      var amount = $( "#products option:selected" ).val();
65      $('#amount').val(amount);
66      console.log($("#products option:selected").text());
67      var description = $( "#products option:selected" ).text();
68      $("#description").val(description);
69      // Open Stripe checkout form
70      stripeFormHandler.open({
71          amount:$('#amount').val() * 100,
72          description:$('#description').val()
73      });
74      e.preventDefault();
75  });
```



Now for the Javascript. The first change is to the checkout-btn click handler. We will go through the changes so do not panic.

Getting the select Element values using JQuery

checkout_ui.js

```
58  /**
59  *   Click event for checkout
60  */
61  $('#checkout-btn').click(function(e){
62      console.log($("#checkout-btn"));
63      console.log($("#products option:selected").val());
64      var amount = $("#products option:selected").val();
65      $('#amount').val(amount);
66      console.log($("#products option:selected").text());
67      var description = $("#products option:selected").text()
68      $("#description").val(description);
69      // Open Stripe checkout form
70      stripeFormHandler.open({
71          amount:$('#amount').val() * 100,
72          description:$('#description').val()
73      });
74      e.preventDefault();
75  });
```

Item: Widgets

ABC \$300 ▾

Checkout

```
Console Search Emulation Rendering
<top frame> Preserve log
Filter [ ] [ ] Regex [All] Errors Warnings Info Logs Debug Handled [x] Hide network messages
2 messages are hidden by filters. Show all messages.
HELLO JQ checkout_ui.js:3
$('#checkout-btn') checkout_ui.js:62
300 checkout_ui.js:63
ABC $300 checkout_ui.js:66
>
```

First we are selecting which product the user choose in the products select element. Here are two console.log statements so you can observe. Remember to remove console.log statements for cross browser testing and production.

Storing Description and Amount in DOM

checkout_ui.js

```
58  /**
59  *   Click event for checkout
60  */
61  $('#checkout-btn').click(function(e){
62      console.log("#checkout-btn");
63      console.log($("#products option:selected").val());
64      var amount = $("#products option:selected").val();
65      $('#amount').val(amount);
66      console.log($("#products option:selected").text());
67      var description = $("#products option:selected").text()
68      $("#description").val(description);
69      // Open Stripe checkout form
70      stripeFormHandler.open({
71          amount: $('#amount').val() * 100,
72          description: $('#description').val()
73      });
74      e.preventDefault();
75  });
```

In the base example for this section we used PHP to store data in the DOM using hidden elements. You can do that also with Javascript. It is really a programmer's choice. In this case in the form using hidden elements seems appropriate.

Checkout Form Description

checkout_ui.js

```
58  /**
59  *   Click event for checkout
60  */
61  $('#checkout-btn').click(function(e){
62      console.log("#checkout-btn");
63      console.log($("#products option:selected").val());
64      var amount = $("#products option:selected").val();
65      $('#amount').val(amount);
66      console.log($("#products option:selected").text());
67      var description = $("#products option:selected").text()
68      $("#description").val(description);
69      // Open Stripe checkout form
70      stripeFormHandler.open({
71          amount:$('#amount').val() * 100,
72          description:$('#description').val()
73      });
74      e.preventDefault();
75  });
```

BEFORE

```
59  /**
60  *   Click event for checkout
61  */
62  $('#checkout-btn').click(function(e){
63      console.log("#checkout-btn");
64      // Open Stripe checkout form
65      stripeFormHandler.open({
66          amount:$('#amount').val() * 100
67      });
68      e.preventDefault();
69  });
```

You use the Stripe Javascript open method to customize the checkout form for each time it opens. We are adding the description here since it will be vary based on user interaction on the form.

Checkout Form Description

BEFORE

```
6  /**
7  *   Create a Stripe configuration object
8  */
9  var stripeConfig = {
10     key:$('#stripe-pk').val(),
11     image:"logo_128x128.png",
12     description:$('#quantity').val() + ' ' + $('#description').val(),
13     //panelLabel:'Order total',
14     name :$("#company-name").val(),
15     //allowRememberMe:false,
```

checkout_ui.js

AFTER

```
6  /**
7  *   Create a Stripe configuration object
8  */
9  var stripeConfig = {
10     key:$('#stripe-pk').val(),
11     image:"logo_128x128.png",
12     //panelLabel:'Order total',
13     name :$("#company-name").val(),
14     //allowRememberMe:false,
```

In the base example we set the Stripe Checkout form description in the configuration data. Now that we are doing that dynamically in the open method, we are removing it here for clarity.

Checkout Form Description

checkout_ui.js

```
58  /**
59  *   Click event for checkout
60  */
61  $('#checkout-btn').click(function(e){
62    console.log("#checkout-btn");
63    console.log($("#products option:selected").val());
64    var amount = $("#products option:selected").val();
65    $('#amount').val(amount);
66    console.log($("#products option:selected").text());
67    var description = $("#products option:selected").text()
68    $("#description").val(description);
69    // Open Stripe checkout form
70    stripeFormHandler.open({
71      amount: $('#amount').val() * 100,
72      description: $('#description').val()
73    });
74    e.preventDefault();
75  });
```



Item: Widgets

ABC \$300 ▾

Checkout



Acme Widgets

Acme Widgets Inc.

ABC \$300

jane@testing.com | [Log out](#)

4242 VISA

09 / 21

Pay \$300.00

Here you see that result from choosing an item on the drop down and it shows in the Stripe Checkout form.

Removed Quantity

BEFORE

```
16 token:function(token){
17     console.log("token", token);
18     dataSend = {};
19     dataSend.stripeToken = token.id;
20     dataSend.amount = $('#amount').val();
21     dataSend.quantity = $('#quantity').val();
22     dataSend.statement_descriptor = $('#statement-descriptor').val();
23     dataSend.description = $('#description').val();
24     dataSend.receipt_email = token.email;
```

checkout_ui.js

AFTER

```
15 token:function(token){
16     console.log("token", token);
17     dataSend = {};
18     dataSend.stripeToken = token.id;
19     dataSend.amount = $('#amount').val();
20     dataSend.statement_descriptor = $('#statement-descriptor').val();
21     dataSend.description = $('#description').val();
22     dataSend.receipt_email = token.email;
```

The quantity is no longer used, so we do not want to send it to our server script. Now it will not appear in the POST data.

Removed Quantity

BEFORE

```
21 // Create a Stripe\Charge OOP object
22 $charge = \Stripe\Charge::create(array(
23     "amount" => $_POST['amount'] * 100, // Convert to cents
24     "currency" => "usd",
25     "source" => $_POST['stripeToken'],
26     "receipt_email" => $_POST['receipt_email'],
27     "statement_descriptor" => substr(strtoupper($_POST['statement_descriptor']), 0, 22)
28     ,
29     "description" => $_POST['quantity'] . ' ' . $_POST['description'])
30 );
```

checkout_charge_card.php

AFTER

```
21 // Create a Stripe\Charge OOP object
22 $charge = \Stripe\Charge::create(array(
23     "amount" => $_POST['amount'] * 100, // Convert to cents
24     "currency" => "usd",
25     "source" => $_POST['stripeToken'],
26     "receipt_email" => $_POST['receipt_email'],
27     "statement_descriptor" => substr(strtoupper($_POST['statement_descriptor']), 0, 22)
28     ,
29     "description" => $_POST['description'])
30 );
```

The quantity key is no longer in the POST data so we removed it.

Sample Test Run

Item: Widgets

ABC \$300 ▾

Checkout



Acme Widgets Inc.
ABC \$300

jane@testing.com | [Log out](#)

••••• 4242 

09 / 21  •••

Pay \$300.00

 **\$300.00 USD** Refund Payment Test

— ch_16edpGGoT2j4xxuJKwjnHjpl

Payment Details Update Description

Amount: \$300.00 USD Statement descriptor: ACME-WIDGETS
Fee: \$9.00 ⓘ
Date: 2015/08/27 23:46:50
Status: Paid ✓
Description: ABC \$300

Card Report fraudulent payment

ID: card_16edpCGoT2j4xxuJACKmmlX Origin: United States 
Name: jane@testing.com
Number: •••••• 4242
Fingerprint: QiM2C5eQ8UmmEhTV
Expires: 9 / 2021
Type: Visa credit card

This is a sample test run.

Wrap Up



```
69 // Open Stripe checkout form
70 stripeFormHandler.open({
71   amount:$('#amount').val() * 100,
72   description:$('#description').val()
73 });
```

Well thats a wrap.

This is just a simple example for a single item checkout but give the user a way to choose the item. Most of the changes are HTML and Javascript using JQuery related.

However, the addition of the dynamic checkout description is particular to our course.

Stripe Checkout Pages Using PHP

Using Stripe JS and AJAX

Bonus: Simple HTML `<select>` Dropdown

With Lon Hosford

© 2015 Alonzo Hosford

Copyright 2015 Alonzo L. Hosford. All Rights Reserved. www.lonhosford.com

This is a Visual Step by Step Workbook and voice transcript for accompanying video for this portion of the course.