

Checkout App Process

Models

Order

Stores delivery fields,
Stripe pid,
Bag items
User_profile

Creates order number
Calculates Total
Overwrites save to set order number

super() gives access to
methods in superclass from a
subclass that inherits it, save
in this case.

Order Line Item

Stores Order id from Order model
Product id from Product model
Sizes
Qty
Line Total

Views

Checkout

Gets form data
Pid with client secret from
template
Json dumps bag
Bag total

Creates Stripe Total
Saves Order

Checkout Success

Gets save_info
Order
If Profile attaches order to it

Deletes Bag
Returns Order to success template for
display

Cache Checkout Data

Gets Pid from form

Adds meta data for save_info box,
Bag (via json dump),
User
to the Stripe Payment Intent (modifies
the payment intent)

Webhooks

Webhook

Path /checkout/wh/ returns the webhook function

Listens for Stripe WebHook
Gets WebHook data & verifies it came from
Stripe

Set up WH_Handler from
webhook_handler.py

Map Webhook events to handler

If there is a webhook handler for the type of
Webhook (payment success or failed) uses it,
otherwise use default

Return to Stripe, the handled event
response (from webhook handler)

Webhook_handler

StripeWH_Handler class

Handle_event
Handles generic or undefined
Stripe Webhook events

Handle_pid_succeeded
Gets pid, bag (from json), save_info,
Billing details
Totals
Saves to profile
Check if order was submitted (by
Checkout)
Saves if order not there

Handle_pid_failed
Return webhook failed received back
to Stripe(via webhook).

Javascript

Checkout.html

Gets Stripe API keys,

Creates card object and mounts it in
checkout template

Handles submitting payment to
Stripe.
Check is save info checkout

POST form to checkout view

POST form to cache_checkout url in
case form not submitted while waiting
for response from stripe

Stripe

