

Stripe Checkout Pages Using PHP

Charging the Card

Overview

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.

Charging the Card



All ready to start charging customers and getting paid?
Well don't starting spending it just yet.

Charging the Card Introduction



We still have a wee bit more work ahead.
So lets get a look at what we need to know and to build for getting paid.

Stripe Checkout Form and Payments

Stripe Checkout Embedded Simple

Item: Widget
Quantity: 12

[Pay with Card](#)

Stripe Checkout Embedded Simple

Acme Widgets Inc.
12 Widgets

Email

Card number

MM / YY CVC

☐ Remember me

[Pay \\$20.00](#)



Stripe Checkout Embedded Simple

Acme Widgets Inc.
12 Widgets

smith@xyz.com

4242 4242 4242 4242

09 / 2021 123

☐ Remember me

[✓](#)

The Stripe Checkout form does the heavy lifting of securing a valid payment source for us. It frees us from worrying about the details of handling or validating payment cards.

Stripe Checkout Form and Payments



But it does not finalize the transaction by collecting a payment. This is because the next step in the checkout process can vary depending on your needs. However in this section we will focus on just getting paid.

Using the Stripe API to Make a Charge

Stripe Checkout Embedded Simple

Item: Widget
Quantity: 12
[Pay with Card](#)

Stripe Checkout Embedded Simple

Acme Widgets Inc.
12 Widgets

Email

Card number

MM / YY CVC

☐ Remember me

[Pay \\$20.00](#)



Stripe Checkout Embedded Simple

Acme Widgets Inc.
12 Widgets

smith@xyz.com

4242 4242 4242 4242

09 / 2021 123

☐ Remember me

[Pay](#)



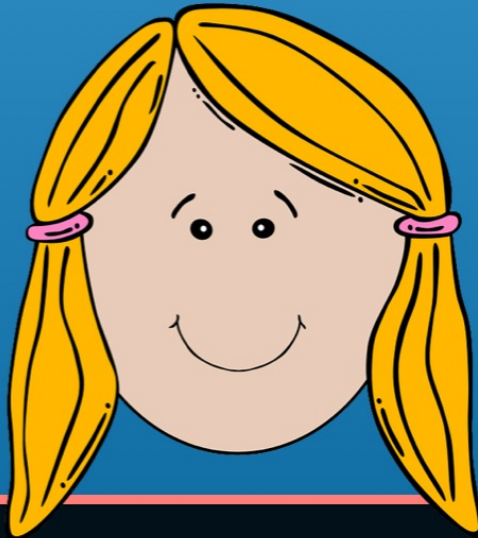
For you to get paid you need to write some additional PHP code.

Using the Stripe API to Make a Charge

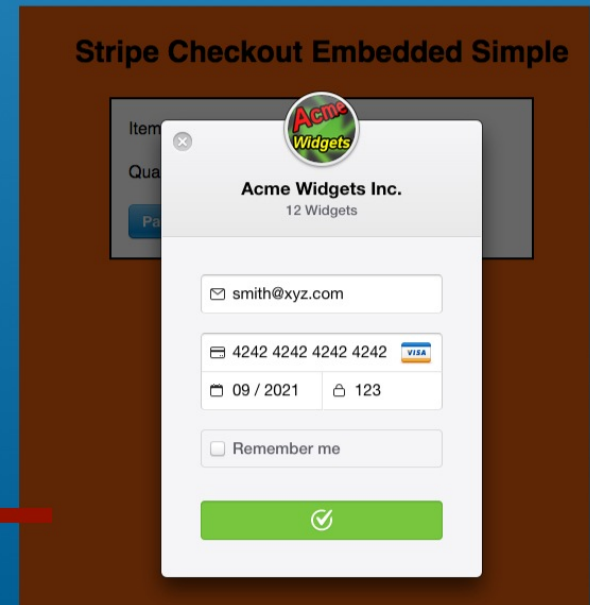


That PHP code will use the Stripe PHP library and communicate with the Stripe API. The Stripe PHP library helps minimize the coding that we need to do.

Charging The Customer

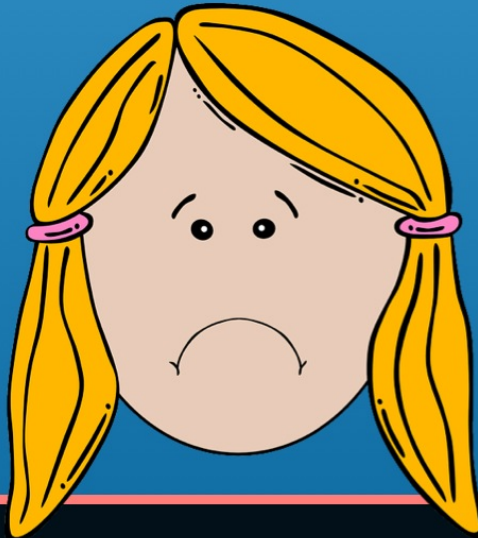


Thank You for Your Order



When we go to collect from customers, we hopefully will get successful results.

Charging The Customer



**Something Horrible Went Wrong!
We are Working to Get it Fixed!**

Card Exception

InvalidRequest Exception

Authentication Exception

APIConnection Exception

Base Exception

Exception



Stripe Checkout Embedded Simple

A screenshot of a Stripe Checkout form. At the top, it says 'Acme Widgets Inc.' and '12 Widgets'. Below that is a text input field for an email address, containing 'smith@xyz.com'. There is a card number input field showing '4242 4242 4242 4242' and a Visa logo. Below the card number is an expiration date field showing '09 / 2021' and a security code field showing '123'. There is a checkbox labeled 'Remember me'. At the bottom is a green button with a white checkmark icon.

But we can get exceptions from Stripe.
Those we will need to learn how to handle.

Charging The Customer

Card Exception

InvalidRequest Exception

Authentication Exception

APIConnection Exception

Base Exception

Exception

stripe api

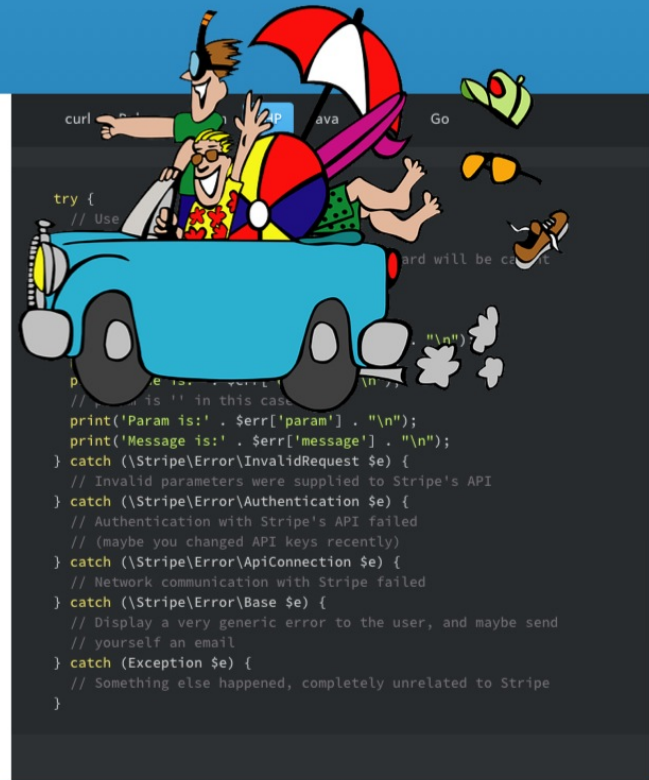
API
Introduction
Authentication
Errors
Pagination
Versioning
Expanding objects
Metadata
Idempotent requ...

METHODS

Charges
Refunds
Customers
Cards
Subscriptions
Plans
Coupons
Discounts


Handling errors

Our API libraries can raise exceptions for many reasons, such as a failed charge, invalid parameters, authentication errors, and network unavailability. We recommend always trying to gracefully handle exceptions from our API.



So we plan to learn more about them using the online documentation. Then we will narrow that down to the coding we need for a our work.

Using the Dashboard



The image displays the Stripe dashboard interface, which is used for managing payments and transactions. The dashboard is divided into several sections:

- Left Sidebar:** Contains navigation links for Dashboard, Customers, Recipients, TRANSACTIONS (Payments, Transfers, Balance), SUBSCRIPTIONS (Plans, Coupons), REQUESTS (Events & Webhooks, Logs), and a Test button.
- Main Content Area:** Displays the details of a POST request to the `/v1/charges` endpoint, dated 2015/03/15 07:29:09.

POST /v1/charges
2015/03/15 07:29:09

Summary

- Time: 2015/03/15 07:29:09
- Method: POST
- URL: /v1/charges
- Status: 200
- IP address: 24.229.160.162
- Version: 2015-02-18 (latest) +
- Source: Stripe/v1 PhpBindings/2.1.1
- Related: Charge — ch_15gZvdIlpyNluc9S19QNFdDt

Parsed Request Query Parameters

No query parameters

Parsed Request POST Body

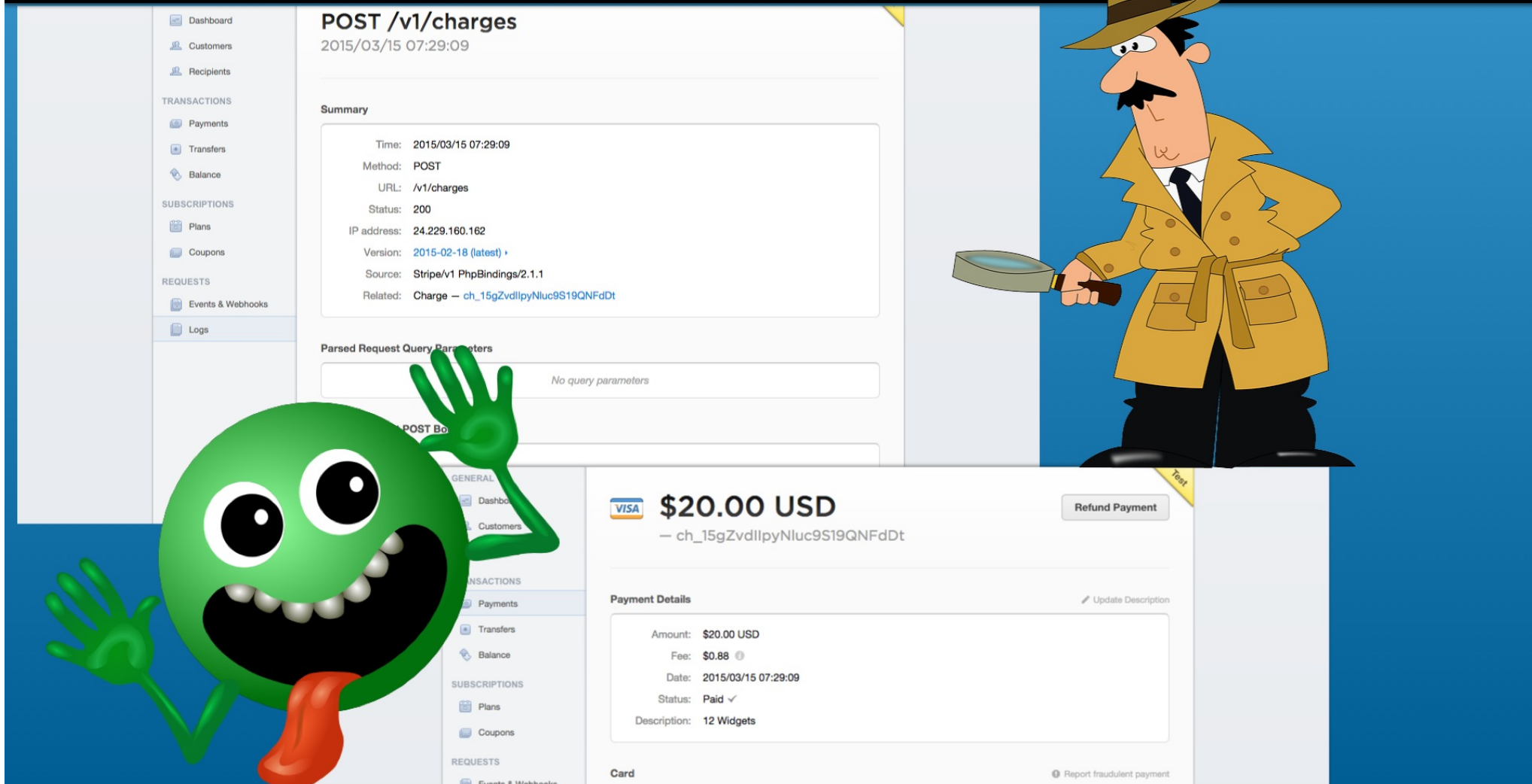
{

The dashboard also shows a detailed view of a payment transaction:

- Payment Details:** Amount: \$20.00 USD, Fee: \$0.88, Date: 2015/03/15 07:29:09, Status: Paid ✓, Description: 12 Widgets.
- Card:** Card details and a link to Report fraudulent payment.
- Refund Payment:** A button to initiate a refund.

We also are going to learn more about what is going on in your Stripe dashboard.

Using the Dashboard



The image is a composite of several elements. On the left, a green alien character with large eyes and a wide, toothy grin is waving. In the background, the Stripe dashboard is visible, showing a sidebar with navigation links like Dashboard, Customers, Recipients, Transactions, Payments, Transfers, Balance, Subscriptions, Plans, Coupons, Requests, Events & Webhooks, and Logs. The main content area displays a 'POST /v1/charges' request log for 2015/03/15 07:29:09. The log includes a 'Summary' section with details like Time, Method, URL, Status, IP address, Version, Source, and Related. Below this is a 'Parsed Request Query Parameters' section showing 'No query parameters'. On the right, a cartoon detective character in a yellow trench coat and hat is holding a magnifying glass. In the foreground, a payment receipt for \$20.00 USD is shown, with details like Amount, Fee, Date, Status, and Description. The receipt also includes a 'Refund Payment' button and a 'Report fraudulent payment' link.

POST /v1/charges
2015/03/15 07:29:09

Summary

Time: 2015/03/15 07:29:09
Method: POST
URL: /v1/charges
Status: 200
IP address: 24.229.160.162
Version: 2015-02-18 (latest) [View](#)
Source: Stripe/v1 PhpBindings/2.1.1
Related: Charge — ch_15gZvdllpyNluc9S19QNFdDt

Parsed Request Query Parameters

No query parameters

Payment Details

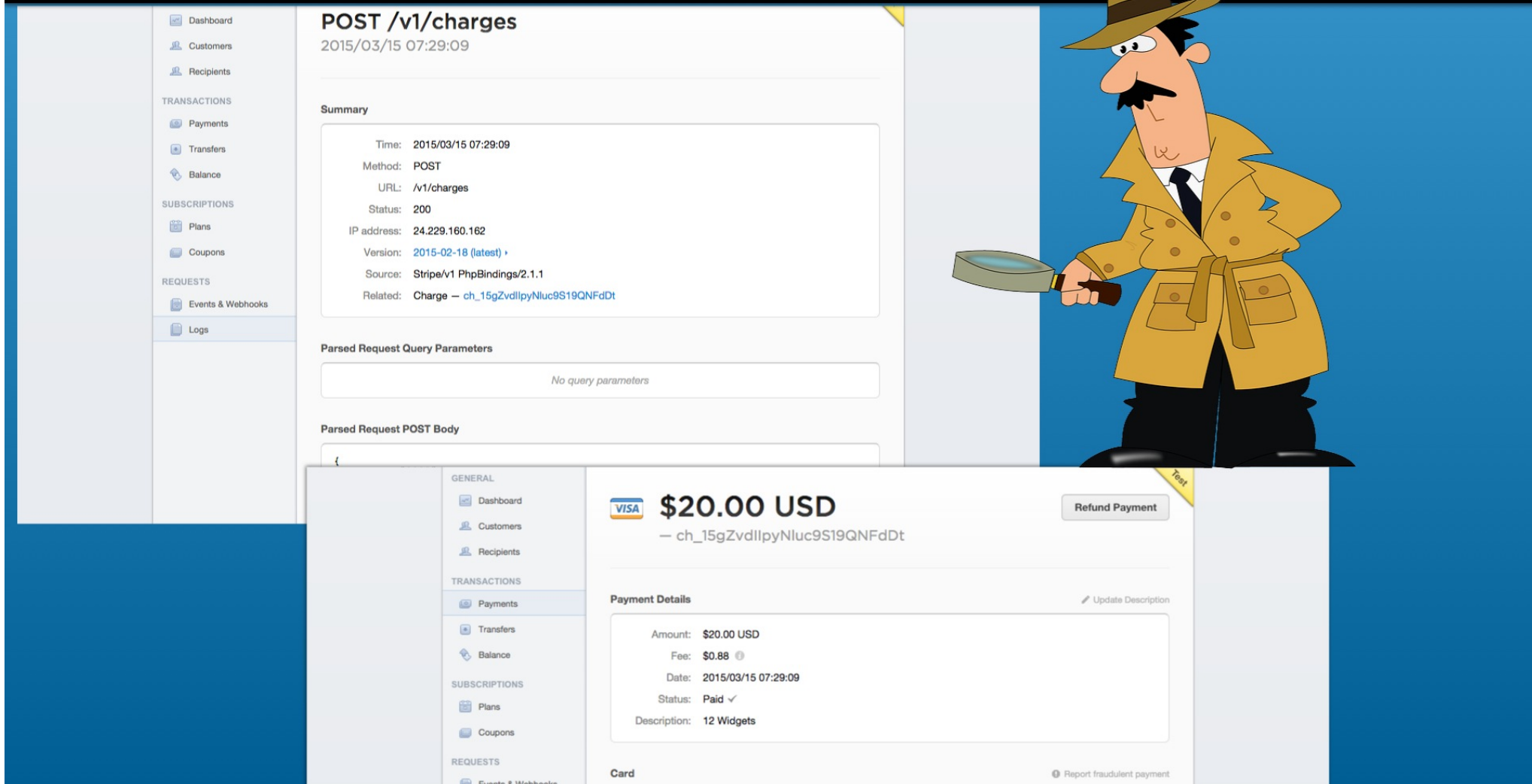
Amount: \$20.00 USD
Fee: \$0.88 [View](#)
Date: 2015/03/15 07:29:09
Status: Paid [View](#)
Description: 12 Widgets

Card

Report fraudulent payment

Don't worry we do not think that is a government auditor.
But Stripe does record all the activity.

Using the Dashboard



POST /v1/charges
2015/03/15 07:29:09

Summary

Time: 2015/03/15 07:29:09
Method: POST
URL: /v1/charges
Status: 200
IP address: 24.229.160.162
Version: 2015-02-18 (latest) [View](#)
Source: Stripe/v1 PhpBindings/2.1.1
Related: Charge — ch_15gZvdllpyNluc9S19QNFdDt

Parsed Request Query Parameters

No query parameters

Parsed Request POST Body

{

GENERAL

TRANSACTIONS

Payments

Payment Details [Update Description](#)

Amount: \$20.00 USD
Fee: \$0.88 [View](#)
Date: 2015/03/15 07:29:09
Status: Paid [View](#)
Description: 12 Widgets



Card [Report fraudulent payment](#)

Refund Payment

Test

This includes successful checkouts, charges to customers and payments to you. You can research all these records as needed. This is helpful in writing code and if customers have inquiries.

Knowing the Stripe Data



stripe api

API
Introduction
Authentication
Errors
Pagination
Versioning
Expanding objects
Metadata
Idempotent requ...

METHODS
Charges
The charge object
Create a charge
Retrieve a charge
Update a charge
Capture a charge
List all charges
Refunds

The charge object

ATTRIBUTES

id	—	string
object	—	string, value is "charge"
livemode	—	boolean
amount	—	Amount charged in cents positive integer or zero
captured	—	If the charge was created without capturing, this boolean represents whether or not it is still uncaptured or has since been captured. boolean
created	—	timestamp

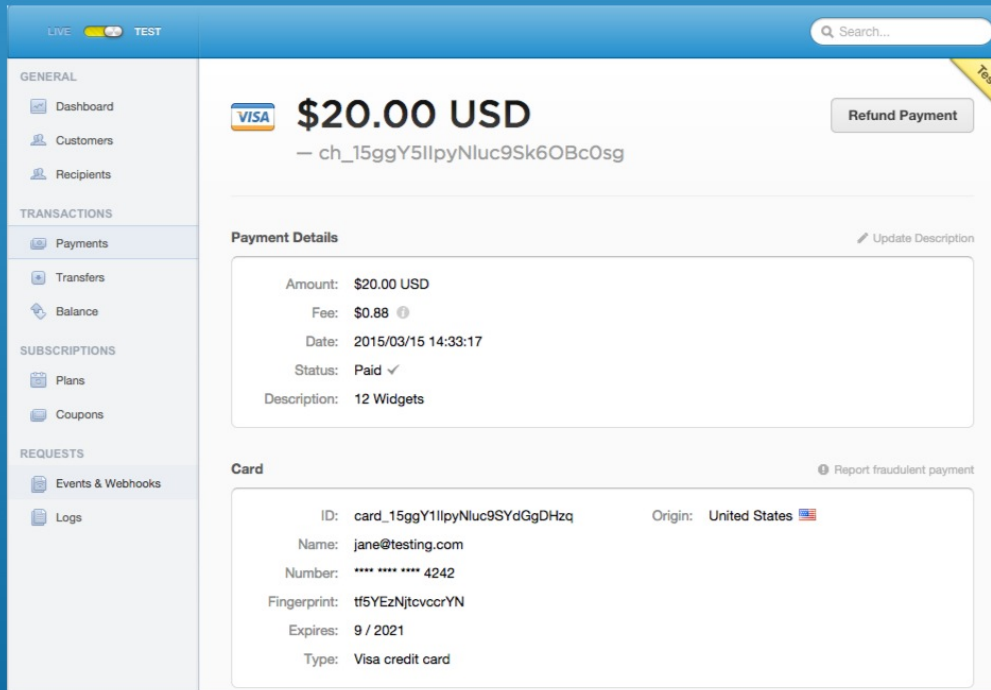
curl Ruby Python **PHP** Java Node Go

Example Response

```
Stripe\Charge JSON: {
  "id": "ch_15ggV5IIpyNluc9Sk60Bc0sg",
  "object": "charge",
  "created": 1426444397,
  "livemode": false,
  "paid": true,
  "status": "succeeded",
  "amount": 2000,
  "currency": "usd",
  "refunded": false,
  "source": {
    "id": "card_15ggV5IIpyNluc9SYdGgDHqz",
    "object": "card",
    "last4": "4242",
    "brand": "Visa",
    "funding": "credit",
    "exp_month": 9,
    "exp_year": 2021,
    "fingerprint": "tf5YEzNjtcvccrYN",
    "country": "US",
    "name": "jane@testing.com",
    "address_line1": null,
    "address_line2": null,
    "address_city": null,
    "address_state": null,
    "address_zip": null,
    "address_country": null,
```

To do that better we will take a look at the information available online.
That helps us understand data that we will see in the dashboard reports.

Taking Dashboard Actions



The screenshot displays the Stripe dashboard interface. At the top, there are tabs for 'LIVE' and 'TEST', and a search bar. The left sidebar contains navigation links for 'GENERAL' (Dashboard, Customers, Recipients), 'TRANSACTIONS' (Payments, Transfers, Balance), 'SUBSCRIPTIONS' (Plans, Coupons), and 'REQUESTS' (Events & Webhooks, Logs). The main content area shows a payment of \$20.00 USD from a Visa card. The payment details include the amount, fee, date, status (Paid), and description. The card details include the ID, name, number, fingerprint, expiration date, and type.

GENERAL

- Dashboard
- Customers
- Recipients

TRANSACTIONS

- Payments
- Transfers
- Balance

SUBSCRIPTIONS

- Plans
- Coupons

REQUESTS

- Events & Webhooks
- Logs

Payment Details

Amount: \$20.00 USD
Fee: \$0.88
Date: 2015/03/15 14:33:17
Status: Paid ✓
Description: 12 Widgets

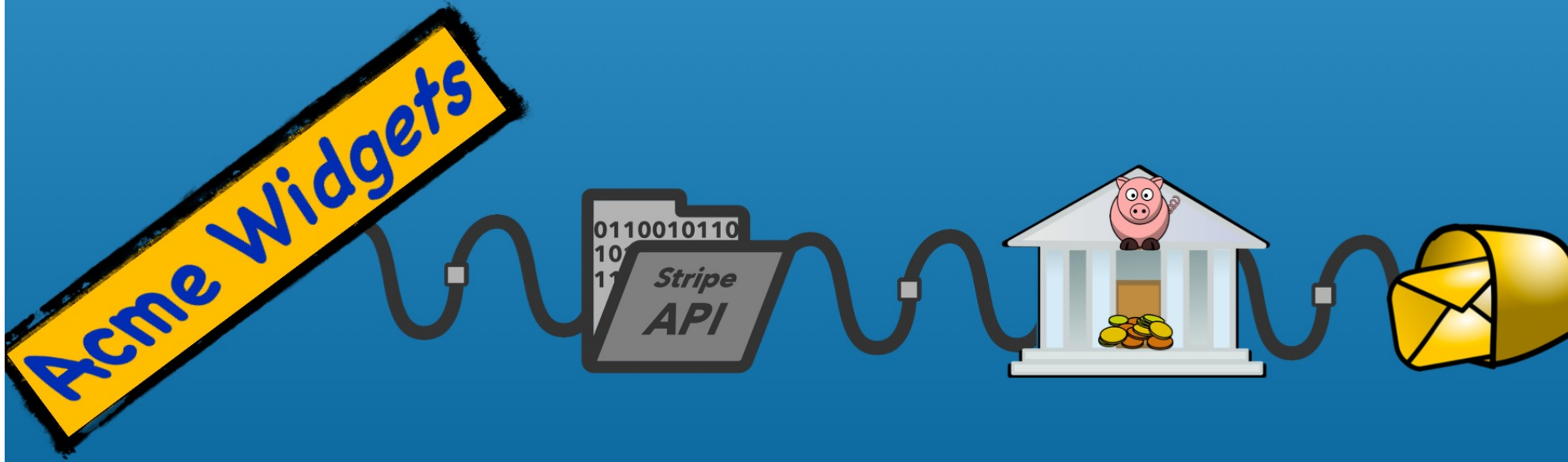
Card

ID: card_15ggY1lpyNluc9SYdGgDHZq
Name: jane@testing.com
Number: **** * 4242
Fingerprint: tf5YEzNjtcvccrYN
Expires: 9 / 2021
Type: Visa credit card



The Stripe dashboard is a convenient back office interface to what happens on your web site. You will find that you can make changes to transactions such as making refunds or reissuing payment notices.

Your Website to Customer's Bank Statements



Finally the order information from your Stripe transactions on your website is sent to the customer's bank. We will look at what input we have to that information to avoid confusing charges on customer bank statements.

Wrap Up



Well that's it!
You can get ready to go.
See you at the first session.

Stripe Checkout Pages Using PHP

Charging the Card

Overview

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.