# Package ‘RStripe’

October 12, 2022

**Title**  A Convenience Interface for the Stripe Payment API

**Version**  0.1

**Author**  Evan Farrell <evandev@gastrograph.com>

**Maintainer**  Evan Farrell <evandev@gastrograph.com>

**Description**  A convenience interface for communicating with the Stripe payment processor to accept payments online. See <https://stripe.com> for more information.

**Depends**  R (>= 3.1.1)

**Imports**  jsonlite, RCurl, httr

**License**  MIT + file LICENSE

**LazyData**  true

**RoxygenNote**  5.0.1

**NeedsCompilation**  no

**Repository**  CRAN

**Date/Publication**  2016-07-15 00:28:26

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>rstripe</td>
<td>3</td>
</tr>
<tr>
<td>stripe_balance_history</td>
<td>3</td>
</tr>
<tr>
<td>stripe_cancel_transfer</td>
<td>4</td>
</tr>
<tr>
<td>stripe_capture_charge</td>
<td>4</td>
</tr>
<tr>
<td>stripe_close_dispute</td>
<td>5</td>
</tr>
<tr>
<td>stripe_create_account_token</td>
<td>5</td>
</tr>
<tr>
<td>stripe_create_application_fee_refund</td>
<td>6</td>
</tr>
<tr>
<td>stripe_create_card</td>
<td>7</td>
</tr>
<tr>
<td>stripe_create_card_token</td>
<td>7</td>
</tr>
<tr>
<td>stripe_create_charge</td>
<td>8</td>
</tr>
<tr>
<td>stripe_create_coupon</td>
<td>9</td>
</tr>
<tr>
<td>stripe_create_customer</td>
<td>10</td>
</tr>
<tr>
<td>stripe_create_invoice</td>
<td>11</td>
</tr>
<tr>
<td>stripe_create_invoice_item</td>
<td>11</td>
</tr>
<tr>
<td>stripe_create_plan</td>
<td>12</td>
</tr>
<tr>
<td>Function</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>stripe_create_recipient</td>
<td>13</td>
</tr>
<tr>
<td>stripe_create_refund</td>
<td>14</td>
</tr>
<tr>
<td>stripe_create_subscription</td>
<td>14</td>
</tr>
<tr>
<td>stripe_create_transfer</td>
<td>15</td>
</tr>
<tr>
<td>stripe_delete_card</td>
<td>16</td>
</tr>
<tr>
<td>stripe_delete_coupon</td>
<td>16</td>
</tr>
<tr>
<td>stripe_delete_customer</td>
<td>17</td>
</tr>
<tr>
<td>stripe_delete_customer_discount</td>
<td>17</td>
</tr>
<tr>
<td>stripe_delete_invoice_item</td>
<td>18</td>
</tr>
<tr>
<td>stripe_delete_plan</td>
<td>18</td>
</tr>
<tr>
<td>stripe_delete_recipient</td>
<td>19</td>
</tr>
<tr>
<td>stripe_delete_subscription</td>
<td>19</td>
</tr>
<tr>
<td>stripe_delete_subscription_discount</td>
<td>20</td>
</tr>
<tr>
<td>stripe_list_application_fees</td>
<td>20</td>
</tr>
<tr>
<td>stripe_list_application_fee_refunds</td>
<td>21</td>
</tr>
<tr>
<td>stripe_list_cards</td>
<td>21</td>
</tr>
<tr>
<td>stripe_list_charges</td>
<td>22</td>
</tr>
<tr>
<td>stripe_list_coupons</td>
<td>23</td>
</tr>
<tr>
<td>stripe_list_customers</td>
<td>23</td>
</tr>
<tr>
<td>stripe_list_events</td>
<td>24</td>
</tr>
<tr>
<td>stripe_list_invoices</td>
<td>25</td>
</tr>
<tr>
<td>stripe_list_invoice_items</td>
<td>25</td>
</tr>
<tr>
<td>stripe_list_plans</td>
<td>26</td>
</tr>
<tr>
<td>stripe_list_recipients</td>
<td>26</td>
</tr>
<tr>
<td>stripe_list_refunds</td>
<td>26</td>
</tr>
<tr>
<td>stripe_list_subscriptions</td>
<td>27</td>
</tr>
<tr>
<td>stripe_list_transfers</td>
<td>27</td>
</tr>
<tr>
<td>stripe_pay_invoice</td>
<td>28</td>
</tr>
<tr>
<td>stripe_retrieve_account</td>
<td>28</td>
</tr>
<tr>
<td>stripe_retrieve_application_fee</td>
<td>29</td>
</tr>
<tr>
<td>stripe_retrieve_application_fee_refund</td>
<td>30</td>
</tr>
<tr>
<td>stripe_retrieve_balance</td>
<td>30</td>
</tr>
<tr>
<td>stripe_retrieve_balance_transaction</td>
<td>31</td>
</tr>
<tr>
<td>stripe_retrieve_card</td>
<td>31</td>
</tr>
<tr>
<td>stripe_retrieve_charge</td>
<td>32</td>
</tr>
<tr>
<td>stripe_retrieve_coupon</td>
<td>32</td>
</tr>
<tr>
<td>stripe_retrieve_customer</td>
<td>33</td>
</tr>
<tr>
<td>stripe_retrieve_event</td>
<td>33</td>
</tr>
<tr>
<td>stripe_retrieve_invoice</td>
<td>34</td>
</tr>
<tr>
<td>stripe_retrieve_invoice_item</td>
<td>34</td>
</tr>
<tr>
<td>stripe_retrieve_invoice_line_items</td>
<td>35</td>
</tr>
<tr>
<td>stripe_retrieve_plan</td>
<td>35</td>
</tr>
<tr>
<td>stripe_retrieve_recipient</td>
<td>36</td>
</tr>
<tr>
<td>stripe_retrieve_refund</td>
<td>36</td>
</tr>
<tr>
<td>stripe_retrieve_subscription</td>
<td>37</td>
</tr>
<tr>
<td>stripe_retrieve_transfer</td>
<td>37</td>
</tr>
<tr>
<td>stripe_upcoming_customer_invoice</td>
<td>38</td>
</tr>
<tr>
<td>stripe_update_application_fee_refund</td>
<td>38</td>
</tr>
</tbody>
</table>
stripe_update_card .......................................................... 39
stripe_update_charge ......................................................... 40
stripe_update_coupon ......................................................... 40
stripe_update_customer ....................................................... 41
stripe_update_dispute ....................................................... 42
stripe_update_invoice ......................................................... 42
stripe_update_invoice_item .................................................. 43
stripe_update_plan ............................................................ 44
stripe_update_recipient ....................................................... 44
stripe_update_refund ........................................................ 45
stripe_update_subscription .................................................. 46
stripe_update_transfer ....................................................... 46

Index 48

rstripe

Description

rstripe.

stripe_balance_history

Retrieve a Stripe Account Balance History.

Description

Retrieve the history of your account balance.

Usage

stripe_balance_history(api_key, args = NULL)

Arguments

api_key  Your Stripe API Key
        optional list which can include
        available_on optional: Filter on date with unix timestamp.
        ending_before optional: An object id which will show objects before
        limit optional: A number 1 to 100 to limit the items. Default is 10
        source optional: Only returns transactions associated with ID.
        transfer optional: Only returns transactions associated with transfer.
        starting_after optional: An object id which will show objects starting here
        type optional: Type: 'charge', 'refund', 'adjustment', 'application_fee' 'application_fee_refund', 'transfer', or 'transfer_failure'
        currency optional: 3 letter ISO
stripe_capture_charge

**Value**
A data frame with balance history information if succeeded.

---

**stripe_cancel_transfer**

*Cancel a transfer on Stripe.*

**Description**
This will cancel a transfer that is scheduled.

**Usage**

```r
stripe_cancel_transfer(api_key, transfer_id)
```

**Arguments**

- `api_key` Your Stripe API Key
- `transfer_id` The id for the tranfer you want to cancel

**Value**
A data frame with nothing if successful.

---

**stripe_capture_charge** *Capture a charge.*

**Description**
As part of the two step charge workflow, capture a charge. Read more about the workflow on the stripe website.

**Usage**

```r
stripe_capture_charge(api_key, charge_id, args = NULL)
```

**Arguments**

- `api_key` Your Stripe API Key
- `charge_id` The identifier of the charge to be captured
- `args` an optional list that can include `amount`, `application_fee` *optional* A fee to add on to a charge. Only Stripe Connect
  `receipt_email` *optional* The email to send receipt to.

**Value**
Returns a charge data frame if the capture succeeded.
stripe_close_dispute

Close a dispute.

Description
Close a dispute which you don’t have evidence to.

Usage
stripe_close_dispute(api_key, charge_id)

Arguments
- api_key: Your Stripe API Key
- charge_id: The identifier of the charge with the dispute.

Value
Returns a dispute data frame if closed.

stripe_create_account_token
Create a Bank Account Token

Description
Create a bank account token to schedule a transfer to a recipient.

Usage
stripe_create_account_token(api_key, args)

Arguments
- api_key: Your Stripe API Key
- args: A list which must contain bank account information including
  - country required: Country bank account is in. Only 'US' supported.
  - routing_number required: routing number in string form. Not wire number.
  - account_number required: Account number for account. Must be checking.

Value
A data frame with the new bank account token info if successful.
Examples

```r
## Not run: stripe_create_account_token("sk_test_BQokikJ0v8iI2HLWgH4olfQ2",
list(card=list(country="US", routing_number="11000000", account_number="000000014343")))
## End(Not run)
```

---

```r
stripe_create_application_fee_refund

Create a refund for an Application Fee.
```

Description

Create a refund for an application fee.

Usage

```r
stripe_create_application_fee_refund(api_key, application_fee_id, args = NULL)
```

Arguments

- `api_key` Your Stripe API Key
- `application_fee_id` The id for the Application Fee you want to refund
- `args` An optional list which can contain an amount to refund
  - `amount` *optional* default is entire fee
  - `metadata` *optional* A list which can contain anything to further describe.

Value

A data frame with refund information

Examples

```r
## Not run: stripe_create_application_fee_refund("sk_test_BQokikJ0v8iI2HLWgH4olfQ2",
"fee_4cb91Lv8PdUnk")
## End(Not run)
```
stripe_create_card

Add a credit card to Stripe.

Description
Add a new credit card to a customer. Set it as default.

Usage
stripe_create_card(api_key, customer_id, args)

Arguments
- api_key: Your Stripe API Key
- customer_id: The id for the customer you want the credit card added to.
- args: A required list of either a token, or a card dictionary. [https://stripe.com/docs/api#create_card](https://stripe.com/docs/api#create_card)

Value
A data frame with card information

Examples
```r
## Not run: stripe_create_card("sk_test_BQokikJ0vB1I2HlWgH4olFQ2", "cus_4cb7HNjNSqqV2H", 
list(card="tok_14SsjM2eZvkY1o2Ct0xZ2A0")
## End(Not run)

## Not run: stripe_create_card("sk_test_BQokikJ0vB1I2HlWgH4olFQ2", "cus_4cb7HNjNSqqV2H", 
list(card=list(number="42424242424242", exp_month="03", exp_year="2018")))
## End(Not run)
```

stripe_create_card_token

Create a Card Token

Description
Create a card token to be used by other functions. You can use this to create a charge, or to attach to a customer.

Usage
stripe_create_card_token(api_key, args)
Arguments
api_key          Your Stripe API Key
args            A list which must contain card information including

    number       required
    cvc          required
    exp_month    required
    exp_year     required
    address_city optional
    address_country optional
    address_line1 optional
    address_line2 optional
    address_state optional
    address_zip  optional
    name         optional
    customer     optional Customer to attach token to.

Value
A data frame with the new card token info if successful.

Examples
## Not run: stripe_create_card_token("sk_test_BQokikJvBiI2H1WgH4olFQ2",
list(card=list(number="4242424242424242", exp_month="03", exp_year="2018")))
## End(Not run)
customer  optional  Either customer or card required.
card  optional  Either customer or card required.
description  optional  The description of the transfer.
capture  optional  True/false to capture charge later.
recipient_email  optional  The email to send receipt to.
metadata  optional  A list which can contain anything to further describe.
statement_description  optional  15 character description to appear on statement.
application_fee  optional  A fee in cents applied to charge.

Value
A data frame with the new charge info if successful.

Examples

## Not run: stripe_create_charge("sk_test_BQokikJ0vBiI2HIWgH4olfQ2",
  list(card="tok_14SsjM2eZvKYlo2Ctdx2AVO", amount=3000, currency="usd",
  metadata=list(order_id="12545")))
## End(Not run)
stripe_create_customer

Create a Customer.

Description

Create a new customer and add information to them.

Usage

```r
stripe_create_customer(api_key, args = NULL)
```

Arguments

- `api_key` Your Stripe API Key
- `args` an optional list which can contain
  - `account_balance` optional In cents starting account balance. Positive means amount to be paid off. Negative is credit used towards charge.
  - `card` optional Either token from Stripe.js or card details.
  - `description` optional The description of the transfer.
  - `metadata` optional A list which can contain anything to further describe.
  - `coupon` optional A coupon code for a discount applied on all charges.
  - `email` optional Customer's email address.
  - `plan` optional A plan id to subscribe the customer to.
  - `quantity` optional The amount of plans to subscribe them to.
  - `trial_end` optional Timestamp for end of trial period. Overrides plan trial.

Value

A data frame with the new Customer info if successful.
**stripe_create_invoice**  
*Create an Invoice.*

**Description**
Create a new invoice for a customer.

**Usage**
```
stripe_create_invoice(api_key, args)
```

**Arguments**
- **api_key**  
  Your Stripe API Key
- **args**  
  A list which must contain
  - **customer** *required* customer id for the invoice.
  - **description** *optional* description of invoice.
  - **application_fee** *optional* fee in cents to apply to invoice.
  - **metadata** *optional* A list which can contain anything to further describe.
  - **subscription** *optional* id of the subscription to invoice.
  - **statement_description** *optional* Goes on credit card statement of customer.

**Value**
A data frame with the new invoice if successful.

---

**stripe_create_invoice_item**  
*Create an Invoice Item.*

**Description**
Create a new invoice item with amount and attach it to a subscription or invoice. If you leave invoice blank it will be added to the next upcoming invoice.

**Usage**
```
stripe_create_invoice_item(api_key, args)
```
stripe_create_plan

Arguments

- **api_key**: Your Stripe API Key
- **args**: A list which must contain
  - **amount required**: In cents how much for item.
  - **currency required**: 3-letter ISO code for currency.
  - **description optional**: A string to attach to item for easy tracking.
  - **metadata optional**: key/value pairs that you can attach to an invoice item.
  - **invoice optional**: Add to this invoice or leave blank for next invoice.
  - **subscription optional**: Add to a specific subscriptions invoice.

Value

A data frame with the new invoice item if successful.

---

stripe_create_plan Create a Plan

Description

Create a new plan on your stripe account.

Usage

```
stripe_create_plan(api_key, args)
```

Arguments

- **api_key**: Your Stripe API Key
- **args**: A list which must contain
  - **id**: The id of the plan to use. e.g. 'gold'
  - **amount required**: The amount in cents to charge.
  - **currency required**: 3-letter ISO code for currency.
  - **interval required**: Specifies billing frequency. day, week month, year.
  - **name required**: The name of plan. Appears on invoice.
  - **metadata optional**: A list which can contain anything to further describe.
  - **trial_period_days optional**: Specifies trial period in day.
  - **interval_count optional**: intervals in billing. e.g. 3 if interval=month
  - **statement_description optional**: String to appear on invoice. Up to 15 characters

Value

A data frame with the new plan info if successful.
stripe_create_recipient

Examples

## Not run: stripe_create_plan("sk_test_BQokikJ0v8i12HlWgH4olfQ2", 
   list(id="example-gold", amount=3000, currency="usd", interval="month"))
## End(Not run)

stripe_create_recipient

Add a recipient to Stripe.

Description

Add a new recipient to your account to schedule transfers and more.

Usage

stripe_create_recipient(api_key, args)

Arguments

api_key       Your Stripe API Key
args          A required list that must contain

   name  required The legal name of recipient.
   type  required put 'individual' or 'corporation'.
   tax_id optional The tax ID either 'SSN' for individual or 'EIN' for corporation
   bank_account optional token or dictionary of details.
   card optional A US Visa/Mastercard debit card to put transfer on.
   email optional Recipients email address.
   description optional Arbitrary string to describe.
   metadata optional key/value for storing information.

Value

A data frame with new recipient information
### stripe_create_refund  
**Create a Refund**

**Description**

Create a new refund on a charge.

**Usage**

```r
stripe_create_refund(api_key, charge_id, args = NULL)
```

**Arguments**

- `api_key`  
  Your Stripe API Key

- `charge_id`  
  The id of the charge to refund.

- `args`  
  A list can contain:

  - `amount`  
    *required*  
    An amount in cents to refund up to remaining charge.

  - `refund_application_fee`  
    *required*  
    true/false if fee should be refunded. If true, proportional amount of refund is refunded.

  - `metadata`  
    *required*  
    Any metadata to include on the refund.

**Value**

A data frame with the new refund info if successful.

### stripe_create_subscription  
**Create a Subscription**

**Description**

Create a new subscription for a customer and add it to them.

**Usage**

```r
stripe_create_subscription(api_key, customer_id, args)
```
stripe_create_transfer

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_key</td>
<td>Your Stripe API Key</td>
</tr>
<tr>
<td>customer_id</td>
<td>The customer id you want to add a subscription to.</td>
</tr>
<tr>
<td>args</td>
<td>A list which includes:</td>
</tr>
<tr>
<td>plan</td>
<td>The plan id</td>
</tr>
<tr>
<td>coupon</td>
<td>A coupon to apply to this subscription.</td>
</tr>
<tr>
<td>trial_end</td>
<td>When the trial ends. Unix timestamp to override plan. You can use now to make it end now.</td>
</tr>
<tr>
<td>card</td>
<td>A token or card dictionary to use instead of default. required if customer has no card.</td>
</tr>
<tr>
<td>quantity</td>
<td>Quantity to place in subscription.</td>
</tr>
<tr>
<td>application_fee_percent</td>
<td>A decimal 1 to 100 for percentage of amount.</td>
</tr>
<tr>
<td>metadata</td>
<td>A list which can contain anything to further describe.</td>
</tr>
</tbody>
</table>

Value

A data frame with the new subscription info if successful.

stripe_create_transfer

Create a transfer to Stripe.

Description

Create a new transfer that will be first marked as pending.

Usage

stripe_create_transfer(api_key, args)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_key</td>
<td>Your Stripe API Key</td>
</tr>
<tr>
<td>args</td>
<td>A list of arguments</td>
</tr>
<tr>
<td>amount</td>
<td>Amount for transfer.</td>
</tr>
<tr>
<td>currency</td>
<td>The 3-letter currency.</td>
</tr>
<tr>
<td>recipient</td>
<td>The recipient id of the transfer.</td>
</tr>
<tr>
<td>description</td>
<td>The description of the transfer.</td>
</tr>
<tr>
<td>metadata</td>
<td>A list which can contain anything to further describe.</td>
</tr>
<tr>
<td>bank_account</td>
<td>The bank to transfer money to. Include this, card, or neither.</td>
</tr>
<tr>
<td>card</td>
<td>The card to transfer the money to. Include either Bank Account Card, or neither.</td>
</tr>
<tr>
<td>statement_description</td>
<td>A 15 character description to appear on statement.</td>
</tr>
</tbody>
</table>
**stripe_delete_card**  
*Delete a credit card on Stripe.*

**Description**  
Delete a credit card off a customer's account.

**Usage**  
```
stripe_delete_card(api_key, customer_id, card_id)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_key</td>
<td>Your Stripe API Key</td>
</tr>
<tr>
<td>customer_id</td>
<td>The id for the customer where the card is.</td>
</tr>
<tr>
<td>card_id</td>
<td>The id for the card you want to delete</td>
</tr>
</tbody>
</table>

**Value**  
A data frame with card information

**stripe_delete_coupon**  
*Delete a coupon.*

**Description**  
Delete a specific coupon from your account.

**Usage**  
```
stripe_delete_coupon(api_key, coupon_id)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_key</td>
<td>Your Stripe API Key</td>
</tr>
<tr>
<td>coupon_id</td>
<td>The coupon you want to delete</td>
</tr>
</tbody>
</table>

**Value**  
A null dataframe if succeeded.
**stripe_delete_customer**

*Delete a Customer.*

**Description**

Remove a customer from your stripe account.

**Usage**

```python
stripe_delete_customer(api_key, customer_id)
```

**Arguments**

- `api_key`: Your Stripe API Key
- `customer_id`: The id of the customer you want to delete

**Value**

A data frame empty if successful.

---

**stripe_delete_customer_discount**

*Delete a customer discount.*

**Description**

Delete a discount from a customer.

**Usage**

```python
stripe_delete_customer_discount(api_key, customer_id)
```

**Arguments**

- `api_key`: Your Stripe API Key
- `customer_id`: The identifier of the customer who has the discount to be deleted.

**Value**

Returns a data frame deleted true if if succeeded.
stripe_delete_invoice_item

*Delete an Invoice Item.*

**Description**

Remove an invoice item from an invoice.

**Usage**

```python
stripe_delete_invoice_item(api_key, item_id)
```

**Arguments**

- **api_key**: Your Stripe API Key
- **item_id**: The invoice item id you want to delete

**Value**

A data frame without errors if successful.

stripe_delete_plan

*Delete a plan on Stripe.*

**Description**

Delete a plan in your system. Only able to do this if you have no subscriptions on it.

**Usage**

```python
stripe_delete_plan(api_key, plan_id)
```

**Arguments**

- **api_key**: Your Stripe API Key
- **plan_id**: The id for the plan you want to delete

**Value**

A data frame with nothing if successful.
**stripe_delete_recipient**

*Delete a recipient on Stripe.*

**Description**

Remove a recipient from your stripe account.

**Usage**

```
stripe_delete_recipient(api_key, recipient_id)
```

**Arguments**

- `api_key`  
  Your Stripe API Key
- `recipient_id`  
  The id for the recipient you want to delete

**Value**

A data frame with recipient information

---

**stripe_delete_subscription**

*Delete a subscription on Stripe.*

**Description**

Delete a subscription on a particular customer.

**Usage**

```
stripe_delete_subscription(api_key, customer_id, subscription_id)
```

**Arguments**

- `api_key`  
  Your Stripe API Key
- `customer_id`  
  The id of the customer with the subscription.
- `subscription_id`  
  The id for the subscription you want to delete

**Value**

A data frame with nothing if successful.
stripe_delete_subscription_discount

Delete a subscription discount.

Description
Delete a discount from a customer's subscription.

Usage
stripe_delete_subscription_discount(api_key, customer_id, subscription_id)

Arguments
- api_key: Your Stripe API Key
- customer_id: The identifier of the customer who has the discount to be deleted.
- subscription_id: The identifier of the subscription.

Value
Returns a data frame deleted true if it succeeded.

stripe_list_application_fees

List all Application Fees.

Description
List all the application fees. Can filter by charge, date, etc.

Usage
stripe_list_application_fees(api_key, args = NULL)

Arguments
- api_key: Your Stripe API Key
- args: optional list that can include:
  - charge: optional Only return fees associated with this charge.
  - ending_before: optional: An object id which will show objects before
  - limit: optional: A number 1 to 100 to limit the items. Default is 10
  - starting_after: optional: An object id which will show objects starting here

Value
A data frame with the application fees you have created that allow you to refund them.
stripe_list_application_fee_refunds

List all refunds for an Application Fee.

Description

List all refunds associated with a specific application fee.

Usage

```r
stripe_list_application_fee_refunds(api_key, application_fee_id, args = NULL)
```

Arguments

- `api_key`:
  Your Stripe API Key
- `application_fee_id`:
  The id for the Application Fee you want to refund
- `args`:
  An optional list which can contain
  - `ending_before` (optional): An object id which will show objects before
  - `limit` (optional): A number 1 to 100 to limit the items. Default is 10
  - `starting_after` (optional): An object id which will show objects starting here

Value

A data frame with application fee refund object if succeeded

Examples

```r
## Not run: stripe_list_application_fee_refunds("sk_test_BQokikJ0vBiZI2HlWgH4olfQ2","fee_4cbC9ilv8PdUnk", list(limit=3))
## End(Not run) #returns 3 refunds
```

stripe_list_cards

List credit card on customers.

Description

List all the credit cards linked to a customer.

Usage

```r
stripe_list_cards(api_key, customer_id, args = NULL)
```
stripe_list_charges

Arguments

api_key          Your Stripe API Key
customer_id      The id for the customer you want to list the cards
args             an optional list that can contain
    ending_before  optional: An object id which will show objects before
    limit          optional: A number 1 to 100 to limit the items. Default is 10
    starting_after optional: An object id which will show objects starting here

Value

A data frame with card information

Description

List all charges conducted in a time frame or on customer. Limits to 10 originally.

Usage

stripe_list_charges(api_key, args = NULL)

Arguments

api_key          Your Stripe API Key
args             an optional list that can include customer,
    customer optional A customer id to show only their charges.
    ending_before optional An id to show the ones before this item.
    limit            optional The limit to show. Can be 1 to 100. Default is 10.
    starting_after  optional An id to show the ones starting after this item.

Value

Returns a data frame with all customer charges. 10 if no limit specified.
**stripe_list_coupons**  
*List all coupons.*

**Description**
List all the coupons currently in your account.

**Usage**
```
stripe_list_coupons(api_key, args = NULL)
```

**Arguments**
- `api_key`  
  Your Stripe API Key
- `args`  
  An optional list which can contain
  - `ending_before` *optional*: An object id which will show objects before
  - `limit` *optional*: A number 1 to 100 to limit the items. Default is 10
  - `starting_after` *optional*: An object id which will show objects starting here

**Value**
A data frame with all the coupons.

**Examples**
```
## Not run: stripe_list_application_fee_refunds("sk_test_BQokikJ0vBi12HlWgH4olfQ2", "fee_4cbC9iLv8PdUnk", list(limit=3))
## End(Not run) #returns 3 refunds
```

---

**stripe_list_customers**  
*Retrieve a list of Customer.*

**Description**
Retrieve a list of all customers. Defaults up to 10.

**Usage**
```
stripe_list_customers(api_key, args = NULL)
```
**stripe_list_events**

Arguments

- **api_key**
  - Your Stripe API Key

- **args**
  - an optional list of arguments. Can include:
    - **ending_before optional**: An object id which will show objects before
    - **limit optional**: A number 1 to 100 to limit the items. Default is 10
    - **starting_after optional**: An object id which will show objects starting here

Value

A data frame with the Customer list if successful.

---

**Usage**

```r
stripe_list_events(api_key, args = NULL)
```

**Arguments**

- **api_key**
  - Your Stripe API Key

- **args**
  - an optional list that can include:
    - **ending_before optional**: An object id which will show objects before
    - **limit optional**: A number 1 to 100 to limit the items. Default is 10
    - **starting_after optional**: An object id which will show objects starting here
    - **type optional** Can contain specific event name or group with '#' wildcard.

Value

Returns an event data frame containing all the events if valid.
stripe_list_invoices

*List Invoices.*

**Description**

List invoices for a particular customer or all customers within a date.

**Usage**

```r
stripe_list_invoices(api_key, args = NULL)
```

**Arguments**

- `api_key`: Your Stripe API Key
- `args`: An optional list that can include:
  - `ending_before`: An object id which will show objects before
  - `limit`: A number 1 to 100 to limit the items. Default is 10
  - `starting_after`: An object id which will show objects starting here

**Value**

A data frame with the invoices if successful.

---

stripe_list_invoice_items

*List all Invoice Item.*

**Description**

List all the invoice items in your account. This can be limited by customer, date, etc.

**Usage**

```r
stripe_list_invoice_items(api_key, args = NULL)
```

**Arguments**

- `api_key`: Your Stripe API Key
- `args`: A list which can contain:
  - `ending_before`: An object id which will show objects before
  - `customer`: The customer ID whose invoice items to return
  - `limit`: A number 1 to 100 to limit the items. Default is 10
  - `starting_after`: An object id which will show objects starting here

**Value**

A data frame with the invoice items if successful.
**stripe_list_plans**  
*List Plans.*

**Description**
List all the plans in your system. It defaults to showing 10 unless you change the limit.

**Usage**
```
stripe_list_plans(api_key, args = NULL)
```

**Arguments**
- `api_key` Your Stripe API Key
- `args` an optional list that can contain:
  - `ending_before` optional: An object id which will show objects before
  - `limit` optional: A number 1 to 100 to limit the items. Default is 10
  - `starting_after` optional: An object id which will show objects starting here

**Value**
A data frame with plan information

---

**stripe_list_recipients**  
*List all current recipients.*

**Description**
List all the current recipients in your account.

**Usage**
```
stripe_list_recipients(api_key, args = NULL)
```

**Arguments**
- `api_key` Your Stripe API Key
- `args` an optional list that can contain:
  - `verified` optional: Show only verified recipients.
  - `ending_before` optional: An object id which will show objects before
  - `limit` optional: A number 1 to 100 to limit the items. Default is 10
  - `starting_after` optional: An object id which will show objects starting here

**Value**
A data frame with recipient’s information
**stripe_list_refunds**  
*List Refunds*

**Description**
List all the refunds that a charge has. You can have multiple refunds per charge until the full amount is reached.

**Usage**
```
stripe_list_refunds(api_key, charge_id, args = NULL)
```

**Arguments**
- `api_key` Your Stripe API Key
- `charge_id` The id of the charge with the refund.
- `args` an optional list that can contain:
  - `ending_before optional`: An object id which will show objects before
  - `limit optional`: A number 1 to 100 to limit the items. Default is 10
  - `starting_after optional`: An object id which will show objects starting here

**Value**
A data frame with refunds information

**stripe_list_subscriptions**  
*List Subscriptions*

**Description**
List all the subscriptions currently acted on a customer.

**Usage**
```
stripe_list_subscriptions(api_key, customer_id, args = NULL)
```

**Arguments**
- `api_key` Your Stripe API Key
- `customer_id` The id of the customer with the subscriptions.
- `args` an optional list that can contain
  - `ending_before optional`: An object id which will show objects before
  - `limit optional`: A number 1 to 100 to limit the items. Default is 10
  - `starting_after optional`: An object id which will show objects starting here
**stripe_list_transfers  List Transfers.**

**Description**
List all the Bank Account transfers for the account.

**Usage**

```r
stripe_list_transfers(api_key, args = NULL)
```

**Arguments**

- `api_key`  Your Stripe API Key
- `args`  an optional list that can contain:
  - `ending_before`  optional: An object id which will show objects before
  - `limit`  optional: A number 1 to 100 to limit the items. Default is 10
  - `recipient`  optional: The id of a recipient to limit to.
  - `starting_after`  optional: An object id which will show objects starting here
  - `status`  optional: transfers with: `pending`, `paid`, `failed`

**Value**
A data frame with transfers information

---

**stripe_pay_invoice  Pay an Invoice.**

**Description**
Pay the invoice right now with customer information.

**Usage**

```r
stripe_pay_invoice(api_key, invoice_id)
```

**Arguments**

- `api_key`  Your Stripe API Key
- `invoice_id`  The invoice id you want to pay

**Value**
A data frame with the invoice if successful.


---

**stripe_retrieve_account**

*Your Stripe Account.*

---

**Description**

Retrieve your account.

**Usage**

`stripe_retrieve_account(api_key)`

**Arguments**

- `api_key`: Your Stripe API key

**Value**

Your account information in a data frame.

**Examples**

```r
## Not run: stripe_retrieve_account("sk_test_2337VDxzNLS332zsso")
```

---

**stripe_retrieve_application_fee**

*Retrieve an Application Fee.*

---

**Description**

Retrieve the data behind a specific application fee.

**Usage**

`stripe_retrieve_application_fee(api_key, application_fee_id)`

**Arguments**

- `api_key`: Your Stripe API Key
- `application_fee_id`: The id for the Application Fee you want to retrieve

**Value**

A data frame with refund app fee information
stripe_retrieve_application_fee_refund

*Retrieve an existing refund for an Application Fee.*

**Description**
Retrieve the information of a refund.

**Usage**

```python
stripe_retrieve_application_fee_refund(api_key, application_fee_id, refund_id)
```

**Arguments**

- `api_key`: Your Stripe API Key
- `application_fee_id`: The id for the Application Fee you want to refund
- `refund_id`: The id for the Refund you want to retrieve

**Value**

A data frame with refund information

---

stripe_retrieve_balance

*Retrieve Stripe Account Balance.*

**Description**
Retrieve the balance of your stripe account.

**Usage**

```python
stripe_retrieve_balance(api_key)
```

**Arguments**

- `api_key`: Your Stripe API Key

**Value**

A data frame with your account balance
**stripe_retrieve_balance_transaction**

*Retrieve a Stripe Account Balance transaction.*

**Description**

Retrieve information about a transaction on your account.

**Usage**

`stripe_retrieve_balance_transaction(api_key, transaction_id)`

**Arguments**

- `api_key`  
  Your Stripe API Key

- `transaction_id`  
  A stripe you want to get information about.

**Value**

A data frame with balance transaction information if succeeded.

---

**stripe_retrieve_card**  

*Retrieve a card from a customer*

**Description**

Retrieve the information about a credit card. Can retrieve last 4 digits.

**Usage**

`stripe_retrieve_card(api_key, card_id, customer_id)`

**Arguments**

- `api_key`  
  Your Stripe API Key

- `card_id`  
  The id for the card you want to retrieve

- `customer_id`  
  The id for the customer where the card is added to.

**Value**

A data frame with card information
**stripe_retrieve_charge**

*Retrieve a charge.*

**Description**

Retrieve all the information about a specific charge.

**Usage**

`stripe_retrieve_charge(api_key, charge_id)`

**Arguments**

- `api_key`: Your Stripe API Key
- `charge_id`: The identifier of the charge to be retrieved

**Value**

Returns a charge data frame if valid.

**Examples**

```r
## Not run: stripe_retrieve_charge("sk_test_BQokikJOvBiI2HlWgh4olfQ2", 
"ch_14TLyd2eZvKYlo2CdryxwrF8")
## End(Not run)
```

---

**stripe_retrieve_coupon**

*Retrieve a coupon.*

**Description**

Retrieve the information about a coupon.

**Usage**

`stripe_retrieve_coupon(api_key, coupon_id)`

**Arguments**

- `api_key`: Your Stripe API Key
- `coupon_id`: The coupon you want to retrieve.

**Value**

A data frame with the coupon information
**stripe_retrieve_customer**  
*Retrieve a Customer.*

**Description**

Retrieve the information about a specific customer using their id.

**Usage**

```
stripe_retrieve_customer(api_key, customer_id)
```

**Arguments**

- **api_key**: Your Stripe API Key
- **customer_id**: The id of the customer you want to retrieve

**Value**

A data frame with the Customer info if successful.

---

**stripe_retrieve_event**  
*Retrieve an event.*

**Description**

Retrieve the information about a specific event.

**Usage**

```
stripe_retrieve_event(api_key, event_id)
```

**Arguments**

- **api_key**: Your Stripe API Key
- **event_id**: The identifier of the event to be retrieved

**Value**

Returns an event data frame if valid.
**stripe_retrieve_invoice**

*Retrieve an Invoice.*

**Description**
Retrieve all the information on a particular invoice.

**Usage**

```
stripe_retrieve_invoice(api_key, invoice_id)
```

**Arguments**

- `api_key` (Your Stripe API Key)
- `invoice_id` (The invoice id you want to retrieve)

**Value**
A data frame with the invoice if successful.

---

**stripe_retrieve_invoice_item**

*Retrieve an Invoice Item.*

**Description**
Retrieve a specific invoice item’s info using it’s id.

**Usage**

```
stripe_retrieve_invoice_item(api_key, item_id)
```

**Arguments**

- `api_key` (Your Stripe API Key)
- `item_id` (The invoice item id you want to retrieve)

**Value**
A data frame with the invoice item if successful.
stripe_retrieve_invoice_line_items

Retrieve Invoice Items.

Description
Retrieve all the items on an invoice.

Usage
stripe_retrieve_invoice_line_items(api_key, invoice_id, args = NULL)

Arguments
- api_key: Your Stripe API Key
- invoice_id: The invoice id with the items you want to retrieve
- args: An optional list that can include
  - **ending_before** *optional*: An object id which will show objects before
  - **limit** *optional*: A number 1 to 100 to limit the items. Default is 10
  - **starting_after** *optional*: An object id which will show objects starting here

Value
A data frame with the invoice items if successful.

stripe_retrieve_plan
Retrieve a Plan

Description
Retrieve the information on a particular plan on your account.

Usage
stripe_retrieve_plan(api_key, plan_id)

Arguments
- api_key: Your Stripe API Key
- plan_id: The id of the plan you want to retrieve

Value
A data frame with the plan info if successful.
**stripe_retrieve_recipient**

*Retrieve information on a recipient.*

**Description**

Retrieve all the information on a recipient.

**Usage**

```python
stripe_retrieve_recipient(api_key, recipient_id)
```

**Arguments**

- `api_key`  
  Your Stripe API Key
- `recipient_id`  
  The id for the recipient you want to retrieve.

**Value**

A data frame with recipient information

---

**stripe_retrieve_refund**

*Retrieve a refund.*

**Description**

Retrieve all the information about a particular refund on a charge.

**Usage**

```python
stripe_retrieve_refund(api_key, charge_id, refund_id)
```

**Arguments**

- `api_key`  
  Your Stripe API Key
- `charge_id`  
  The id of the charge with the refund.
- `refund_id`  
  The id of the refund you want to retrieve

**Value**

A data frame with the refund info if successful.
**stripe_retrieve_subscription**

*Retrieve a Subscription*

**Description**

Retrieve all the information on a subscription for a customer.

**Usage**

```python
stripe_retrieve_subscription(api_key, customer_id, subscription_id)
```

**Arguments**

- `api_key` Your Stripe API Key
- `customer_id` The id of the customer with the subscription.
- `subscription_id` The id of the subscription you want to retrieve

**Value**

A data frame with the subscription info if successful.

---

**stripe_retrieve_transfer**

*Retrieve a transfer*

**Description**

You can get all the information about a pending transfer or an accepted transfer.

**Usage**

```python
stripe_retrieve_transfer(api_key, transfer_id)
```

**Arguments**

- `api_key` Your Stripe API Key
- `transfer_id` The id of the transfer you want to retrieve

**Value**

A data frame with the transfer info if successful.
stripe_upcoming_customer_invoice

Retrieve upcoming customer’s invoice.

Description
Retrieve the upcoming customer invoice to send out.

Usage

```
stripe_upcoming_customer_invoice(api_key, customer_id)
```

Arguments
- **api_key**: Your Stripe API Key
- **customer_id**: The customer id that you want the upcoming invoice

Value
A data frame with the upcoming invoice if successful.

-------------------

stripe_update_application_fee_refund

Update a refund for an Application Fee.

Description
Update all the information of a refund of an application fee.

Usage

```
stripe_update_application_fee_refund(api_key, application_fee_id, refund_id, args)
```

Arguments
- **api_key**: Your Stripe API Key
- **application_fee_id**: The id for the Application Fee you want to refund
- **refund_id**: The id for the Refund you want to update
- **args**: An optional list which can contain:
  - **metadata**: Optional A list which can contain anything to further describe.
**stripe_update_card**

Update a credit card on Stripe.

**Description**

Update information about the billing address for a credit card as well as other fields.

**Usage**

```r
stripe_update_card(api_key, customer_id, card_id, args)
```

**Arguments**

- **api_key**: Your Stripe API Key
- **customer_id**: The id for the customer you want the credit card added to.
- **card_id**: The id for the card you want to update
- **args**: A list that can contain the following
  - **address_city**: optional
  - **address_country**: optional
  - **address_line1**: optional
  - **address_line2**: optional
  - **address_state**: optional
  - **address_zip**: optional
  - **exp_month**: optional
  - **exp_year**: optional
  - **name**: optional

**Value**

A data frame with card information

**Examples**

```r
## Not run: stripe_create_application_fee_refund("sk_test_BQokikJ0vBiIl2HlWgH4olfQ2", "fee_4cbC9iLv8PdUnk", list(metadata=list(reason="Accidental charge")))
## End(Not run)
```

```r
## Not run: stripe_update_card("sk_test_BQokikJ0vBiIl2HlWgH4olfQ2", "cus_4cb7HNjNSqqV2H", "card_14TLyd2eZvKYloc2C9PXnFmi", list(exp_year="04"))
## End(Not run)
```
stripe_update_charge  Update a charge.

Description
Update the information about a charge.

Usage
stripe_update_charge(api_key, charge_id, args)

Arguments
api_key  Your Stripe API Key
charge_id  The identifier of the charge to be updated
args  A list that can include:

  metadata  optional  A list of metadata to include
  description  optional  A string to describe it

Value
Returns a charge data frame if succeeded.

Examples
## Not run: stripe_update_charge("sk_test_BQokikJ0vBiI2H1WgH4o1fQ2",
"ch_14TLyd2eZvKYoZ2CdryxwF8", list(description="for the item",
metadata=list(order_id="3453")))
## End(Not run)

stripe_update_coupon  Update a coupon.

Description
Update the information about a coupon.

Usage
stripe_update_coupon(api_key, coupon_id, args)
**stripe_update_customer**

**Description**
Update the customer’s information or add a new card to a customer.

**Usage**

```r
stripe_update_customer(api_key, customer_id, args)
```

**Arguments**
- **api_key**: Your Stripe API Key
- **customer_id**: The id of the customer you want to update
- **args**: an optional list which can contain:
  - **account_balance optional**: In cents starting account balance. Positive means amount to be paid off. Negative is credit used towards charge.
  - **card optional**: Either token from Stripe.js or card details.
  - **description optional**: The description of the transfer.
  - **default_card optional**: The id of a card to use as default.
  - **metadata optional**: A list which can contain anything to further describe.
  - **coupon optional**: A coupon code for a discount applied on all charges.
  - **email optional**: Customer’s email address.

**Value**
A data frame with the updated Customer info if successful.
### stripe_update_dispute

*Update a dispute.*

**Description**

Update a dispute and add evidence to it.

**Usage**

```r
stripe_update_dispute(api_key, charge_id, args)
```

**Arguments**

- **api_key**
  - Your Stripe API Key
- **charge_id**
  - The identifier of the charge with the dispute.
- **args**
  - An optional list that can contain
    - **evidence** *(optional)* Evidence towards resolving dispute.
    - **metadata** *(optional)* key/value for storing dispute information.

**Value**

Returns a dispute data frame if valid.

**Examples**

```r
## Not run: stripe_update_dispute("sk_test_BQokikJ0vBiI2HlWgH4olfQ2", "ch_14TLyd2eZvKYlo2Cdryxwrf8", list(evidence="Here's evidence showing this charge is legitimate.",metadata=list(exhibitA="This happened")))
## End(Not run)
```

### stripe_update_invoice

*Update an Invoice.*

**Description**

Update the information on an invoice, add a fee, close, or forgive it.

**Usage**

```r
stripe_update_invoice(api_key, invoice_id, args)
```
**Arguments**

- **api_key**
  - Your Stripe API Key

- **invoice_id**
  - The invoice id you want to update

- **args**
  - Can include:
    - application_fee
    - description
    - closed
    - forgiven
    - statement_description
    - metadata

**Value**

A data frame with the updated invoice if successful.

---

**stripe_update_invoice_item**

*Update an Invoice Item.*

**Description**

Update the information or amount of an invoice item.

**Usage**

`stripe_update_invoice_item(api_key, item_id, args)`

**Arguments**

- **api_key**
  - Your Stripe API Key

- **item_id**
  - The invoice item id you want to update

- **args**
  - Can include
    - **amount** *optional* Amount in cents for item.
    - **description** *optional* A string to attach to item for easy tracking.
    - **metadata** *optional* key/value pairs that you can attach to an invoice item.

**Value**

A data frame with the updated invoice item if successful.
stripe_update_plan  

*Update a Plan*

**Description**

Update the information about a particular plan on your stripe account.

**Usage**

```python
stripe_update_plan(api_key, plan_id, args)
```

**Arguments**

- `api_key`: Your Stripe API Key
- `plan_id`: The id of the plan you want to update
- `args`: A list can contain
  - `name  optional`: The name of plan. Appears on invoice.
  - `metadata  optional`: A list which can contain anything to further describe.
  - `statement_description  optional`: String to appear on invoice. Up to 15 characters

**Value**

A data frame with the updated plan info if successful.

---

stripe_update_recipient  

*Update a recipient on Stripe.*

**Description**

Update or add information about a recipient on stripe.

**Usage**

```python
stripe_update_recipient(api_key, recipient_id, args)
```
**stripe_update_refund**  
*Update a Refund*

**Description**  
Update a refund with any metadata you want to include about the refund.

**Usage**  
```python  
stripe_update_refund(api_key, charge_id, refund_id, args)  
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_key</td>
<td>Your Stripe API Key</td>
</tr>
<tr>
<td>charge_id</td>
<td>The id of the charge with the refund.</td>
</tr>
<tr>
<td>refund_id</td>
<td>The id of the refund you want to update</td>
</tr>
</tbody>
</table>
| args      | A list can contain:

- **metadata optional** Any metadata in a list to change for refund.

**Value**

A data frame with the updated refund info if successful.
stripe_update_subscription

*Update a subscription.*

**Description**

Update the information for a subscription on a customer.

**Usage**

```python
stripe_update_subscription(api_key, customer_id, subscription_id, args)
```

**Arguments**

- **api_key**
  - Your Stripe API Key
- **customer_id**
  - The id of the customer with the subscription.
- **subscription_id**
  - The id of the subscription you want to update.
- **args**
  - which includes:
    - **plan** *required*
      - The plan id
    - **coupon** *optional*
      - A coupon to apply to this subscription.
    - **prorate** *optional*
      - Flag whether to prorate plan when switched. defaults: true.
    - **trial_end** *optional*
      - When the trial ends. Unix timestamp to override plan. You can use `now` to make it end now.
    - **card** *optional*
      - A token or card dictionary to use instead of default. *required* if customer has no card.
    - **quantity** *optional*
      - Quantity to place in subscription.
    - **application_fee_percent** *optional*
      - A decimal 1 to 100 for percentage of amount.
    - **metadata** *optional*
      - A list which can contain anything to further describe.

**Value**

A data frame with the updated subscription info if successful.

stripe_update_transfer

*Update a Transfer*

**Description**

You can update a transfer that is scheduled or has been accepted by changing description.
stripe_update_transfer

Usage

```python
stripe_update_transfer(api_key, transfer_id, args)
```

Arguments

- **api_key**: Your Stripe API Key
- **transfer_id**: The id of the transfer you want to update
- **args**: A list can contain
  - **metadata**: Optional A list of metadata to include
  - **description**: Optional A string to describe the transfer

Value

A data frame with the updated transfer info if successful.
Index

rstripe, 3
rstripe-package (rstripe), 3

stripe_balance_history, 3
stripe_cancel_transfer, 4
stripe_capture_charge, 4
stripe_close_dispute, 5
stripe_create_account_token, 5
stripe_create_application_fee_refund, 6
stripe_create_card, 7
stripe_create_card_token, 7
stripe_create_charge, 8
stripe_create_coupon, 9
stripe_create_customer, 10
stripe_create_invoice, 11
stripe_create_invoice_item, 11
stripe_create_plan, 12
stripe_create_recipient, 13
stripe_create_refund, 14
stripe_create_subscription, 14
stripe_create_transfer, 15
stripe_delete_card, 16
stripe_delete_coupon, 16
stripe_delete_customer, 17
stripe_delete_customer_discount, 17
stripe_delete_invoice_item, 18
stripe_delete_plan, 18
stripe_delete_recipient, 19
stripe_delete_subscription, 19
stripe_delete_subscription_discount, 20
stripe_list_application_fee_refunds, 21
stripe_list_application_fees, 20
stripe_list_cards, 21
stripe_list_charges, 22
stripe_list_coupons, 23
stripe_list_customers, 23
stripe_list_events, 24

stripe_list_invoice_items, 25
stripe_list_invoices, 25
stripe_list_plans, 26
stripe_list_recipients, 26
stripe_list_refunds, 27
stripe_list_subscriptions, 27
stripe_list_transfers, 28
stripe_pay_invoice, 28
stripe_retrieve_application_fee, 29
stripe_retrieve_application_fee_refund, 30
stripe_retrieve_balance, 30
stripe_retrieve_balance_transaction, 31
stripe_retrieve_card, 31
stripe_retrieve_charge, 32
stripe_retrieve_coupon, 32
stripe_retrieve_customer, 33
stripe_retrieve_event, 33
stripe_retrieve_invoice, 34
stripe_retrieve_invoice_item, 34
stripe_retrieve_invoice_line_items, 35
stripe_retrieve_plan, 35
stripe_retrieve_recipient, 36
stripe_retrieve_refund, 36
stripe_retrieve_subscription, 37
stripe_retrieve_transfer, 37
stripe_upcoming_customer_invoice, 38
stripe_update_application_fee_refund, 38
stripe_update_card, 39
stripe_update_charge, 40
stripe_update_coupon, 40
stripe_update_customer, 41
stripe_update_dispute, 42
stripe_update_invoice, 42
stripe_update_invoice_item, 43
stripe_update_plan, 44
INDEX

stripe_update_recipient, 44
stripe_update_refund, 45
stripe_update_subscription, 46
stripe_update_transfer, 46