

Custom Payment Scenarios

Using the API



A little about me...

- *Senior Developer at Skvare*
- *Using CiviCRM since 2010*
- *Reformed Joomla Evangelist*
- *Nitpicker of lesser-used APIs*



Case Study:

Texas Pediatric Society

Case Study: TPS Event Registration

TPS used their event registration to opportunity to make a donation.

- **Pro:** Take advantage of one purchase to make an impulse donation
- **Con:** Difficult to segregate donation from event fee for reporting.

Case Study: TPS Event Registration

Client requirements

- Separate event fee from donation amount
- No redirects or additional pages

Taking the Items Out

1. Decide on financial types or other characteristics of your line items to filter on.

Pro Tip: Use the `FinancialType` API to get the `financial_type_id(s)` by name.

Taking the Items Out

2. Create a preProcess hook
3. Grab the `_params` and `_lineItem` arrays out the form object

Both of these arrays will need to be grabbed using `getVar()`.

Taking the Items Out

4. Loop through the line items for your initial transaction and zero out the line items you'll be transferring.
5. Remove the aggregate of those items from the total.
6. Write the updated items back to the form.

Creating the New Contribution

When the ThankYou form is accessed:

1. Get the `_params` and `_lineItem` arrays again
2. Set a default for the total contribution amount

Creating the New Contribution

When the ThankYou form is accessed:

3. Loop through the line items again and add the line item amount to the contribution total.

Create the transact params array

financial_type_id: Looking for label name

total_amount: use the variable created earlier

contact_id: can be grabbed from the form values

is_test: (optional)

payment_processor: use the numeric id

credit_card_number: \$params[0]['credit_card_number'],

cvv2: \$params[0]['cvv2'],

month: \$params[0]['credit_card_exp_date']['M'],

credit_card_type: \$params[0]['credit_card_type'],

year: \$params[0]['credit_card_exp_date']['Y'],

invoice_id: md5(uniqid(rand(), TRUE)),

skipLineItem: (optional = 1 to use only the total, 0 to keep the line items)

source: (optional)

Creating the New Contribution

- Get the line items via the API with the specified `entity_id` and Financial Type.
- Loop through the line items and update these items with the new `contribution_id` created from the `transact()` call.

Send a Receipt for the New Contribution

- Send the receipt via `contribution.sendconfirmation`

```
$confirmation = civicrm_api3('Contribution', 'sendconfirmation', array(  
  'id' => contribution_id,  
  'receipt_from_email' => "email@example.com",  
  'receipt_from_name' => "From Name",  
));
```

Applications

- Transactions outside of CiviCRM
- Custom transaction workflows for Events, Tax vs. Non-Taxed Contributions, Memberships

Additional Payment APIs in 4.7

Payment

- Adds a payment to an existing contribution
- Does not invoke a payment processor.

Order

- Can create contributions with line items of different types
- Does not invoke a payment processor.

Questions?

Learn More

Questions? Contact Us

Peter Petrik - CEO peter@skvare.com

Mark Hanna - Architect / Senior Developer mark@skvare.com

Jeremy Proffitt - Senior Developer jeremy@skvare.com / @jproffitt

<https://skvare.com> | @skvare