Promo Code (Part 2)

Time id: 1733

Simple Goal: Make this public ready for simple promotion codes. Either percentage or dollars off of a single item or multiple items. Only one promotion code per cart or ecommerce order. If any deeper, it needs to go round 3.  
  
1. JSON to table

Currently storing in the custom\_json table: corp\_id, table\_name (promo\_code\_whole or promo\_code\_percent), main\_id (1 or a real part id number), app\_type\_id (11 is for parts), sub\_id (currently nothing or 0), json\_value with details (see below).

{ "Amount": "1",  
 "StartDate": "3/31/20",  
"EndDate": "4/7/20",  
"PromoCode": "makeitless1"}

Maybe a list of id numbers (what parts it applies to)

Ideas for a new promotion\_codes table – shared table

promotion\_code\_id – key field – auto number – int(10)  
corp\_id – int(10) default 1  
promotion\_code\_name – varchar(25) default null  
promotion\_type – varchar(25) default null – values of “percent”, “dollars off”, “??? – maybe more”  
promotion\_amount – decimal(7,2) default 0.00 – (hold the percent or dollars off value)  
promotion\_code\_notes varchar(255) default null  
promotion\_creation\_date date default 1900-01-01  
promotion\_created\_by\_id int(10) default 1 (payee id of who created it)  
promotion\_status – tinyInt(1) default 1 – 1=active and 0=inactive

See below for the history table  
  
  
2. Link/Button (on cart?) – How do we get to this thing? We don’t have a problem with a link from the cart but it may be tied to a permission.  
  
3. Move code to main pages (not just eXPO)  
  
4. Add dates/time (to/from)  
  
5. Setting (turn on/off Admin level?, tie in with other permissions?)

We are thinking that it would tie-in really well with the customer loyalty points stuff. This deals with master tables, corp tables, and transaction tables. Lots of built-in stuff.  
  
6. Verbiage change (dollars off).  
  
7. Edit promo codes

8. Promotion code through the URL and the FORM scope

9. Be able to search or filter the list of promo codes

- by status, by name, by type, by dates (future)

10. Do we need a history on this?

Yes, we think that we do need a history on this.

New history table would be called promotion\_code\_history – shared table

promotion\_code\_history\_id – int(10) – auto number  
corp\_id int(10) default 1  
promotion\_code\_id int(10) default 1  
payee\_id int(10) default 1 (who did it – the history record)  
promotion\_history\_flag\_id int(10) default 1  
promotion\_history\_date\_time datetime 1900-01-01 12:00:00  
promotion\_history\_note varchar(150) default null

We need to record a history every time things get created, edited, changed, or set to active/inactive.

11. Tech specs – coding things

- When we use a promotion code, we need to add it to the line (if known) in the campaign field.

- Externally, we are using standalone discounts… what about internally on the shopping cart (secure)? In-line or standalone discounts? We get better math if we can use in-line discounts and taxes work out better.

- Internal usage??? What needs to happen there? What about custom carts? Cascade settings where needed? We will need a new page to show the history for the promotion codes (per promotion code level).

- Ecommerce usage??? What needs to happen there?