Promo Code (Part 2)

Time id: 1733

Start date 4/1/20 to 4/15/20

Rough sketch of the project. More detail is needed. Just the rough draft with concept ideas.

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)

As of 4/9/20 we thought that using the existing discounts table would be better than creating a brand new table from scratch. See below on number 12.

Ideas for a new promotion\_codes table – shared table – once again, see number 12 for ideas on using the discounts table.

If we built our own… it may look something like this…

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

We see an advantage of getting this info (custom json stuff) into a table sooner than later. This may even be a sub project to get it out the initial json stuff.  
  
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)

See the number 5 option for turning it on and setting it up. We know that eXPO has this built out in their ecommerce package. We may need to go in and grab so of their code and tweak it to our needs.

Also, on the internal (secure) point of sale, if you make a promo code, it shows up as an option. We need to rewire that based on the special account settings and rules.

We also need a way to get to the add/edit promotion codes stuff. Number 5 may provide that avenue. Put it under special accounts.

We currently have a custom interface that allows for discounts to be added and edited. If we use that same table, we may be able to use that interface for the add/edit promotion code or add/edit coupon stuff.  
  
4. Add dates/time (to/from)

We decided to use the existing discounts table to handle days, weeks, and start/end date/times. If we need more about the discount engine stuff, here is some light research.

* News and updates
  + https://www.adilas.biz/new\_and\_updates/2019/04/05/new-discount-engine-coming-soon/ - new discount engine released
  + https://www.adilas.biz/new\_and\_updates/2019/11/20/how-to-set-the-discount-engine/ - how to set the discount engine
* Do some looking around and light discovery on what is there and what we want.
* We know there is a permission – id=164 – Set Manager Discounts – Discount Engine
* Known homepage - https://www.adilas.biz/top\_secret/custom/discount.cfm

Help file - https://www.adilas.biz/top\_secret/help.cfm?id=516&pwd=discount  
  
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 (special accounts). This deals with master tables, corp tables, and transaction tables. Lots of built-in stuff. This already has permissions and so does the discount engine stuff.

Along those lines, we decided to help build a history table for the discounts (auto discounts, promo codes, and coupons). See below for more info.  
  
6. Verbiage change (dollars off).

The current promo code uses percent and whole as the types. We were thinking that “percent” or “dollars off”. If we use the discounts table, it used 1’s and 0’s.  
  
7. Edit promo codes

Currently, we can’t edit the promo codes. It is either build a new one or delete. If we go the discount way, we will need to be able to edit the promotion codes. That may already be built in.

8. Promotion code through the URL and the FORM scope

The URL and FORM scope deals with how promotion codes get passed to the system. Currently, the only scope or format is the form scope. The URL scope would allow for ads, buttons, and other links to be clicked. The code just needs to check for either one and then flip it into the session or cart info structure.

We have tons of samples of codes that do the flip/flop between FORM and URL or URL to FORM. See the top\_secret/secure/advanced\_invoice\_search\_results.cfm file for some scope flipping.

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

- by status, by name, by type, by dates (future) – wants for the current one.

If we go with the discount table, we should be able to filter them already.

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 discount\_history – shared table

discount\_history\_id – int(10) – auto number  
corp\_id int(10) default 1  
discount\_id int(10) default 1  
payee\_id int(10) default 1 (who did it – the history record)  
discount\_history\_flag\_id int(10) default 1  
discount\_history\_date\_time datetime 1900-01-01 12:00:00  
discount\_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?

12. Future stuff

As a fun side note, the existing discount engine has a number of these type of fields and ideas that are listed below. See the table called “discounts”. As we looked at it, it would cover almost all of the things listed below. We also talked about adding in some new fields that would allow us to declare if the entry was an automatic discount (normal for that table) or a promotion code, or a coupon. We also had some discussions about the in-line discounts vs standalone discounts. Currently, the discounts table is automated and only does in-line discounts. We also want notes, tied out to history tables, Great potential there.

If we decide to use the discounts table… here are the new fields that we would want and need.

discount\_notes varchar(255) default null  
discount\_creation\_date date default 1900-01-01  
discount\_created\_by\_id int(10) default 1 (payee id of who created it)  
 - this may be the payee\_id in the discounts table, we didn’t check that yet  
discount\_category varchar(25) default “auto discount” – this is where we could hold things like “promo code”, “coupon”, etc.  
discount\_display\_function varchar(25) default “in-line” – this is where we could set the discount type to be “in-line” or “standalone”.

- dates/times  
- days of the week  
- time of the day  
- lists of items  
- limit by part category  
- limit by vendor  
- limit by customer type  
- exclusions – everything except…  
- usage counts, how many times it may be used  
- usage per person or per corp (max number of times)  
- by one get one or combo packs or packages  
- min and maxes – must buy x to get something off or up to max number of items – window  
- barcode the promotion code – either URL or a scan or from a web app or ad (magazine, newspaper, paper flyer, etc.)  
- coupon type options  
- expiration date