

Points Redemption in Decimals

Setup and Functionality	2
Enabling Decimal Redemption and Redemption Strategies	2
Redemption Strategies	2
Functionality when Decimal Redemption Configuration is Enabled	3
View on Membercare and API	3
IsRedeemable and Redeem APIs	5
Points Communication Tags	6
Current Behavior	6
Inconsistency with API/Membercare and Upcoming Changes	6
New Tags	6
Round-off to Floor for Existing Tags	7
Tags that will Display Actual Decimals	7
Points Currency Communication Tags	7
Current Behavior	7
Recently Released Feature - Decimals for Base Currency	7
Upcoming Changes - floor and New Tags	8

Setup and Functionality

Loyalty now supports redemption of points in decimals. This enables use cases like complete payment of bill using points e.g. \$ 4.95 bill paid through points. This also benefits loyalty programs that have high points-to-currency ratio.

Enabling Decimal Redemption and Redemption Strategies

By default the points redemption in decimals is **disabled** for programs, so there's no change in existing redemption behavior.

The points redemption in decimals could be enabled from Organization Settings (Miscellaneous > Points Management). The configuration is titled 'Allow Redemption in Decimal Places'

Organisation settings | A Animesh Org

redemption |

Miscellaneous > Points Management
Redemption Configuration

Submit

Thanks for your interest in redeeming. Your redemption validation code is {{validation_code}}

Please select from these tags: {{validation_code}}|{{lifetime_points}}|{{lifetime_purchases}}|{{loy}}|{{store_name}}

Redemption Configurations

VALIDATION	VALIDATION
REDEMPTION_VALIDATION_REQUIRED	<input type="checkbox"/>
ALLOW REDEMPTION IN DECIMAL PLACES	<input checked="" type="checkbox"/>
ALLOW SKIPPING VALIDATION	<input type="checkbox"/>
IS STORE ID PART OF VALIDATION CODE	<input type="checkbox"/>
INCLUDE POINTS IN LOYALTY VALIDATION	<input type="checkbox"/>
CONF_CLIENT_POINTS_REDEMPTION_TRANSACTION_NUMBER_REQUIRED	<input checked="" type="checkbox"/>
OTP VALIDATION FOR POINTS TRANSFER DISABLED	<input checked="" type="checkbox"/>

Redemption Strategies

Once the redemption in decimal is enabled, redemption strategies have to be modified/created against individual programs to support redemption of points in decimal

Edit redemption settings

Name * 3 Decimals Redemption

Same for all tiers ON

Allow redemptions in multiple of * 0.001

Minimum Points to be redeemed * 0.001

Maximum Points that can be redeemed * 1000

Minimum lifetime points required * 1

Minimum lifetime purchases required * 0

Minimum current points required * 1

Description

Save Cancel

The redemption strategies have to be modified/created to support decimal redemption because till now the minimum value of 'Allow redemptions in multiple of' is 1. So with the existing redemption strategies, redemption will happen in integers only.

With the redemption in decimal configuration enabled, the minimum value of 'Allow redemptions in multiple of' will depend on the number of decimals supported by the program. (The number of decimals supported by a program is defined by 'Round Decimal' configuration in Points Allocation strategy.) So if a program supports 2 decimals, the minimum redemption value will be 0.01; if the program supports 3 decimals, the minimum redemption value will be 0.001.

Functionality when Decimal Redemption Configuration is Enabled

View on Membercare and API

When points redemption in decimal is enabled, the GET APIs that display customers' lifetime points, current points, redeemed points, points on a transaction, etc. will display points in decimal. (The number of decimal places will be the same as the number of decimal places supported by the program.)

Similarly, Membercare displays customers' points related KPIs in decimal places based on the number of decimal places supported by the program.

[Table of Contents](#)

Note that some of the existing APIs already supported points in 2 decimal places, so those APIs will still display points in decimals even if the points redemption in decimal configuration isn't enabled.

A Animesh Org?⚙️👤

Loyalty Information

Test and Control status	TEST	Preferred Store	N/A
Loyalty Type	LOYALTY		
Select Loyalty Program	BUKL	Tier	Basic
Lifetime purchases	11883	<u>Current points</u>	<u>2794.724</u>
<u>Lifetime points</u>	<u>2801.722</u>	<u>Redeemed points</u>	<u>6.998</u>
Promised points	0	Expired points	0
Returned points	0	Tier expiry date	2120-04-29 23:59:59
Adjusted points	0		

Subscription Status

More Information

TransactionsPointsCouponsInteractionsBehavioural EventsLeadsOthers

PurchasesReturns

Show 10 entriesSearch:

Date	Store name	Transaction id	Transaction number	Amount	Discount	Issued points	Promised points	Returned points	Net points	Outlier status	Program Name
06 May 2020 13:53:18	ANIMESHWEB	38247177	Bill-2020-05-06-06	699	0	Multiple Programs	Multiple Programs	Multiple Programs	352.891	NORMAL	Multi Program
06 May 2020 13:52:18	ANIMESHWEB	38247176	Bill-2020-05-06-05	699	0	Multiple Programs	Multiple Programs	Multiple Programs	352.891	NORMAL	Multi Program
06 May 2020 13:51:18	ANIMESHWEB	38247175	Bill-2020-05-06-04	699	0	Multiple Programs	Multiple Programs	Multiple Programs	352.891	NORMAL	Multi Program
06 May 2020 13:50:18	ANIMESHWEB	38247173	Bill-2020-05-06-03	699	0	Multiple Programs	Multiple Programs	Multiple Programs	352.891	NORMAL	Multi Program

1. BUKL : 200.123
2. BU Fashions : 152.768

More Information

Transactions Points Coupons Interactions Behavioural Events Leads Others

Redemptions Expiry schedule Expired points Slab history Promotions Loyalty Tracker

Show 10 entries

Date	Store name	Transaction Number	Points Redeemed	Points Reversed
30 Apr 2020 11:32:11	animesh_2	BUKL130420-04	2.998	
30 Apr 2020 10:43:11	animesh_2	BUKL130420-02	1.9	
30 Apr 2020 10:35:11	animesh_2	BUKL130420-01	2.1	

Showing 1 to 3 of 3 entries

IsRedeemable and Redeem APIs

In the current scenario when points redemption in decimal is disabled, the redemption API allows decimal as input. However, the API internally truncates the decimal value and redeems the whole number e.g. if 2.34 points is passed for redemption, the redemption happens for 2 points only

When points redemption in decimal is enabled, both IsRedeemable and Redeem APIs will support points redemption in decimal without truncating the decimals to the whole number. The number of decimal places supported in redemption will be the same as the number of decimals supported by the program. (The number of decimals supported by a program is defined by 'Round Decimal' configuration in Points Allocation strategy.)

E.g. if a program supports two decimals and minimum redemption value in redemption strategy is 0.01, the points could be redeemed in the multiple of 0.01 (2.01, 5.02, 8.03, and 8.99).

In this case, however, if decimals passed in IsRedeemable and Redeemable APIs are more than the number supported by the program, the APIs will give error. In the above example, if a program supports two decimals and somebody tries to redeem points in three decimals e.g. 9.123, they will get an error.

Also, a new error message is added when the points to be redeemed isn't in the multiple of minimum redemption value in the redemption strategy. E.g. if the minimum value in the redemption strategy is 0.05, and somebody tries to redeem 2.34 points, they will get an error that the points to be redeemed isn't divisible by the minimum redemption value.

Points Communication Tags

Current Behavior

The existing points tag including current points, lifetime points, points on event, etc. display points information in decimals. The number of decimals in the tags is the same as the number of decimals supported for the program.

If the value of one of the tags has more decimals than that supported for the program, the decimal gets **rounded-off** for most of the tags. E.g. if a program supports 1 decimal and a customer has current points as **200.34**, the current points tag will have the value **200.3**; in the same program, if another customer has current points as **400.35**, their current points tag will have the value **400.4**

Inconsistency with API/Membership and Upcoming Changes

While in loyalty communication, most of the points tag are **rounded-off**, API/Membership uses **floor** for points KPIs. This creates inconsistency in points information across different modules

E.g. for a program that supports 1 decimal for points and has decimal redemption **enabled**, **400.35** current points will be displayed as **400.3** on Membership/API and **400.4** in points communication.

Similarly for a program that supports 1 decimal for points has decimal redemption **disabled**, **400.35** current points will be displayed as **400** on Membership/API and **400.4** in points communication.

To address this inconsistency, we are making the following changes in the current/next sprints.

- Add new tags for displaying points value in whole numbers (to support programs that don't use decimal redemption)
- Change behavior of existing points tags from **round-off** to **floor**
- Redeemed points/transferred points tags will display actual redeemed decimal irrespective of the configuration

New Tags

- Loyalty Points Int
- Lifetime Points Int
- Points on Event Int
- Points Expiring Int

The above tags will display points in whole numbers with **floor** configuration. E.g. in case of **400.35** current points, Loyalty Points Int tag will display **400**

[Table of Contents](#)

Tags that will Display Actual Decimals

- Redeemed Points
- Points Transferred
- Points Reversed

Round-off to Floor for Existing Tags

- Remaining points tags including
 - Current Points (aka Loyalty Points)
 - Lifetime Points
 - Points on Event
 - Points Getting Expired

Points Currency Communication Tags

Current Behavior

The existing points currency tags including Current Points Currency, Lifetime Points Currency, Points on Event Currency, and Redeemed Points Currency include **2 decimals**

Recently Released Feature - Decimals for Base Currency

A new configuration has been added to define the number of decimal places for base currency. This configuration is currently used only for the above mentioned points currency tags

So for an Org, if the number of decimal places for base currency is set as **3**, the tags current points currency, lifetime points currency, points on event currency, and redeemed points currency in **3 decimals**.

If decimal places for the base currency isn't defined, the default decimal places for the above points currency tags will continue to be **2**. So existing behavior won't get impacted

The new configuration on decimal places for base currency is available on Org Currency Setup screen (https://nightly.capillary.in/org/setup/Profile?flash=&step_name=org_currency)

Table of Contents

Organization Profile > Set > **Set Org Currency** > Organization Languages/Countries > P: > O > Social

Organization Currency Configuration

Albanian lek (L)
Alderney pound (€)
Algerian dinar (دج)
Angolan kwanza (Kz)
Argentine peso (\$)
Armenian dram (AMD)
Aruban florin (f)
Ascension pound (€)
Australian dollar (\$)
Bahamian dollar (\$)

United Arab Emirates dirham (د.إ.)

Currency Available: add >> << remove

Base Currency: United Arab Emirates dirham (د.إ.)

Decimal Places for Base Currency: 3

Submit

Upcoming Changes - *floor* and New Tags

To make the loyalty tags consistent, the points currency tags will also use *floor* instead of round-off.

E.g. for an Org with decimal places for base currency as **1**, if the currency value of current points for a customer is **400.35**, the current points currency tag will get resolved to **400.3**

In addition, new tags will be introduced to support whole number values for currency tags. These new tags include:

- Loyalty Points (in \$) Int
- Lifetime Points (in \$) Int
- Points Expiring (in \$) Int
- Points on Event (in \$) Int