**>> How records get added into table 'wahed\_rebalnce' ?**

1. When cron runs we check whether previous data exist in table 'wahed\_changein\_nav' for each user.

2. If old data exist we calculate ChangeInAmount as new['DepositsWithdrawals'] minus prev['DepositsWithdrawals'].

3. If old data does not esist then ChangeInAmount will be 0.

4. We calculate TnxType as if new['DepositsWithdrawals'] > prev['DepositsWithdrawals'] then TnxType = 'Deposite' and new['DepositsWithdrawals'] < prev['DepositsWithdrawals'] then TnxType = 'Withdrawal';

5. We add ClientAccountID, DepositsWithdrawals(getting from xml), ChangeInAmount, LastUpdated, UpdatedDate and TnxType into table 'wahed\_rebalnce'.

**>> Trading > Lineup - Trade lineup table**

1. Users withdrwal request are save into table 'wahed\_funds\_transactions'. Get Withdrawals details from the table 'wahed\_funds\_transactions' whose TransactionType = 'Withdrawal' and IsViewed = 1.

2. Get actual deposites details from table 'wahed\_rebalnce' where ChangeInAmount > 0 and IsViewed = 0

3. These withdrwal and deposites details are merged and shown in Trade lineup table.

4. Column UserModel displays current portfolio of the user.

5. Users current 'MarketValue' is calculated as sum(EndingValue - Commissions), 'TotalGain' calculated as Sum(Mtm + Interest + Dividends + FxTranslation + AdvisorFees) from table 'wahed\_changein\_nav' using latest updated values form cron.

6. Column 'Trades' / 'View Trade'

a. Basically it needs three parameters TnxType, TnxAmt(Deposite or Withdrwal amount), ClientAccountID and user's current model(Risk profile).

b. Get user current model ID. Get details of all active users, using this list get details of required user and his current model ID. [$userCurrentModel]

c. Get user's current MarketValue using ClientAccountID.

d. Calculate RebalanceAmt. For Withdrawals Rebalance Amount equal to current Market Value minus Withdrawal amount. For Deposits Rebalance Amount equal to current Market Value. [$rebalanceAmt]

d. If TnxType is 'Withdrawal' then RebalanceAmt = CurrMV - TnxAmt else if TnxType is 'Deposit' then RebalanceAmt = CurrMV.

e. Get user’s positions details from table ‘wahed\_open\_positions’ its values are calculated as below [$positionsData]

|  |  |
| --- | --- |
| ClientAccountID | Available in table |
| Description | Available in table |
| Symbol | Available in table |
| Quantity | Available in table |
| FXRateToBase | Available in table |
| CurrentAmt | Quantity \* CostBasisPrice \* FXRateToBase |
| CostBasisPrice | CostBasisPrice \* FXRateToBase |
| MarkPrice | MarkPrice \* FXRateToBase |
| Position Value @Base | PositionValue \* FXRateToBase |
| Unrealized G/L | FifoPnlUnrealized \* FXRateToBase |

f. Get user’s latest cash details from table ‘wahed\_nav’ as shown below [$cashData]

|  |  |
| --- | --- |
| Cash | 124.757748759 |
| Stock | 5738.16 |

g. Get user’s model details using current model id. Depending on his current model or Risk profile or Portfolio we will come to know about percentage allocation for each security (ex. shown below). Data is fetched from tables ‘wahed\_models\_list’ and ‘wahed\_model\_securities’. [$model\_details]

|  |  |
| --- | --- |
| ISEM | 5 |
| ISUS | 6 |
| ISWD | 5 |
| PHAUl | 6 |
| M1GU | 0 |
| PSA | 15 |
| CASH | 63 |

h. Creating new array which list all securities with its 'Position Value @Base' and cash with its value. (Ex. shown below)

|  |  |
| --- | --- |
| Position | Position Value @Base |
| ISEM | 131.15 |
| ISUS | 325.76 |
| ISWD | 574.04 |
| PHAUl | 467.36 |
| M1GU | 0 |
| WISEX | 4254.31 |
| PSA | 1283.95 |
| CASH | 215.3 |

i. Now we have list of all positions and its value. We need to calculate percentage allocation of each security and cash if we consider sum of ‘Position Value @Base’ of all securities and CASH as 100%. Ex. shown below[$percentageNewArr]

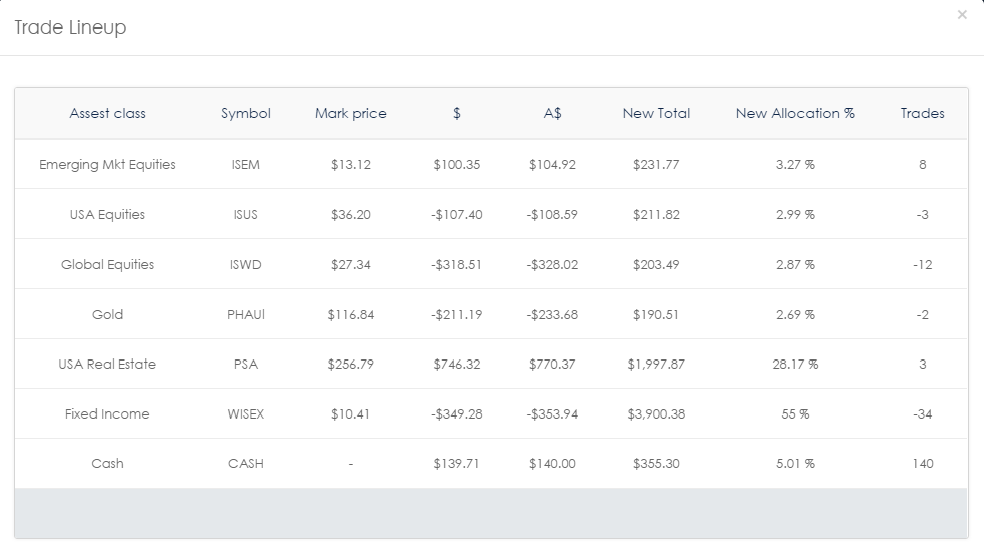
|  |  |
| --- | --- |
| ISEM | 2.37 |
| ISUS | 11.62 |
| ISWD | 20.06 |
| PHAUl | 18.26 |
| M1GU | 0 |
| PSA | 45.57 |
| CASH | 2.13 |

j. The array values $positionsData, $cashData, $percentageNewArr, $userCurrentModel, $model\_details and $rebalanceAmt are used for further calculations.

k. Now calculate $currentTotalAmt using $positionsData and $cashData.

It is calculated as sum of ‘CurrentAmt’ of all positions user is holding PLUS ‘cash’ user have.

l. Lets understand table – “Trade Lineup”

****

**COLUMN – Asset Class**

Every security is assigned an Asset class which is defined in file customcfg.php

**COLUMN - Symbol**

Each security has unique Symbol

**COLUMN – Mark Price**

Each security has Market Price calculated as MarkPrice \* FXRateToBase which we are getting in XML, We have Mark Price for each security in $positionsData array.

**COLUMN - $ and Trades**

We are calculating this column values for **each security** using helper function amtToBuySellDoller(). This function returns two values “dollerValue” and “trade”.

This function accepts following five parameters.

|  |  |
| --- | --- |
| Parameter name | How its calculated |
| $totalCurrentAmt | You can describe it as total amount of user’s portfolio.  Calculated as follows –  Total Amount of all security (calculated as: Quantity \* CostBasisPrice \* FXRateToBase) + Cash value. |
| $amtAddingWithdrawing | This is the amount user wants to add or withdraw.  Also known as rebalance amount. |
| $targetAllcationEachSec | Target allocation of security as per User Model or Portfolio. |
| $CurrentSecurityAmt | For Individual Security:  Amount of each security (Quantity \* CostBasisPrice \* FXRateToBase).  For Cash:  Current amount of cash |
| $MarkPrice | For Individual Security:  Current market price of security (MarkPrice \* FXRateToBase).  For Cash: Current market price of cash  = 1 |

**$** = ((($totalCurrentAmt + $amtAddingWithdrawing) \* ($targetAllcationEachSec / 100)) - $CurrentSecurityAmt);

**Trade** = round (**$** / $MarkPrice );

**Column - A$**

For Individual Security:

Calculated as Trade (calculated in above step) \* $MarkPrice.

A$ = Trade \* $MarkPrice

For Cash:

Calculated as Trade (calculated in above step) \* $MarkPriceCash.

A$ = Trade \* $MarkPriceCash (for cash $MarkPriceCash = 1)

**Column - New Total**

For Security:

$CurrentAmt (from $positionsData) + A$(Calculated in above step for security).

For Cash

Cash value (from $cashData) + A$(Calculated in above step for Cash).