**File changes for the** Refiltering/Reprocessing

**resources/views/charts.blade.php**  
  
Global Object containing the filters :  
  
globalObj -> filtersObj {

dailyLossLimit : <Boolean>,

weeklyLossLimit : <Boolean>,

maxDailyLosers : <Boolean>,

maxConsecutiveDailyLoss : <Boolean>,

maxConsecutiveWeeklyLoss : <Boolean>

};

Purpose => Object will let us know which limits filters are applicable on the current page.

gloabalObj -> currentFilters [

{FIRST\_FORM\_FILTERS},

{WHAT\_IF\_FORM\_FILTERS}

];

Purpose => Global object will contain the previously saved filters by the user, or the default filters in case no filters were saved previously.  
  
Html Development : HTML forms for the filters and What IF Filters.  
  
**public/assets/js/pages/chartapp.js**

saveFilters() => New Method  
  
Trigger Event => User click on the refresh icon against any filter.  
  
Purpose => Updates the filters for the users in the member table.  
  
Procedure :

* read the filter and What If form
* Convert the values to Array of Object  
  [{FIRST\_FORM\_FILTERS}, {WHAT\_IF\_FORM\_FILTERS}]
* generate request to the [ZoomChartController@saveFilters](mailto:ZoomChartController@saveFilters)() to update the members table.
* Update the gloabalObj -> currentFilters.

**app/Http/Controllers/ZoomChartController.php**

1. saveFilters() => New Method  
   Purpose => Updates the member table `basic\_settings` and `advanced\_settings` columns  
     
   Procedure :

Receives the array of Object [{FIRST\_FORM\_FILTERS}, {WHAT\_IF\_FORM\_FILTERS}]

updates the two columns with the respective values.

1. Index() => Existing Method

Purpose => create the json object for the already saved filter settings by the user in case they have been saved  
If not present create the default object and returns the same to the charts.blade.php view.

**public/assets/js/pages/chartapp.js**checkFilters() => New Method  
  
Inputs => ChartType <String>  
Returns => <Object> Filters object that can be appended to the extra filters for the charts

Procedure :

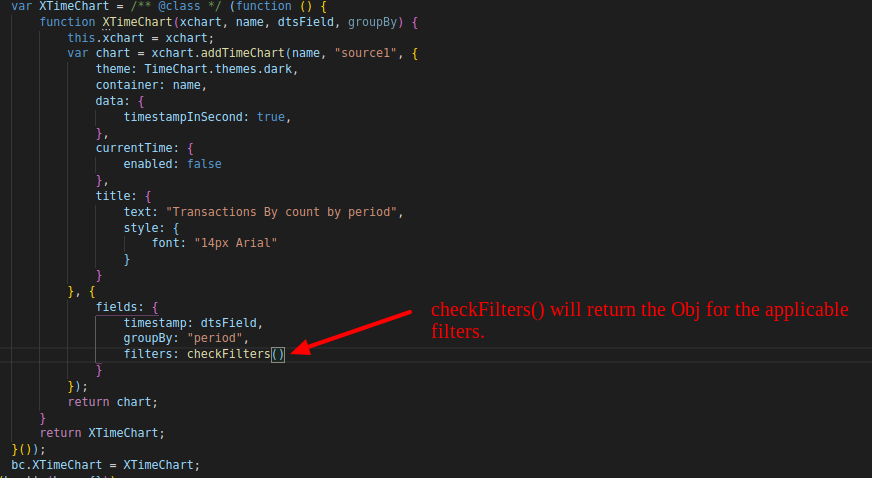
* read the gloabalObj -> currentFilters object.
* Generate Obj for the filter option present in currentFilters as

return Obj filters [{

type : ‘jx’ => need to make the change on the xcharts so that it ignores these filters  
 field : ‘dailyLossLimit’,

}]

* return the filters Obj

The function checkFilters() will be called as in the below image for each type of Filter. For example in case of XtimeCharts it will be called as in below :

Now , XtimeChart Knows which Filter it have to handle.  
  
  
**public/assets/js/pages/xchart.js**

functions that need to be changed :

1. matchFilter  
     
   Line Number 902 => while reading the extraFilters it should ignore filters whose type is ‘jx’   
     
   Condition should say => if (filter.type == ‘js’) continue;
2. createTimeChartResponse  
     
   Inputs => chartId, chart, params  
     
   Procedure :  
      
    read the extraFilters from the Parameter params.  
      
    If filters obj is not empty  
     
   var \_return = JXFilters() \\on **xchart\_filters.js – New File**If (\_return) include row else continue;

**public/assets/js/pages/xchart\_filters.js – New File**

JXFilters()  
  
This will include the procedure for filters like daily loss limit and others.  
  
**INPROGRESS FROM HERE.**