

Calculations

Project has been accepted and we'll invest significant time on this in coming weeks / months . What is currently called [Advanced Rating](#) syntax should be renamed to Tiki **Calculations** which can be used all over Tiki, including advanced ratings and the [Mathematical Calculation Tracker Field](#)

This page is to brainstorm and plan to make Tiki better at calculations. It should be possible for any data in Tiki to be used for calculations and for the results to be used anywhere in Tiki, to do all kinds of useful things: alerts, reports, etc.

Overview

- [Trackers](#) are used to collaborate on data input. This part is great. Many field types, flexible permissions, good reporting, etc.
- [PluginTrackerStat](#) can be useful but is not configurable enough
- [Mathematical Calculation Tracker Field](#) offers a lot of power and flexibility on various data fields of the same tracker item (like rows in a spreadsheet) and do things like [Ease Importance Priority](#) , with a diversity of syntax: [comment](#), [Multiply](#), [Divide](#), [Sum](#), [Substract](#), [round](#), [coalesce](#), [str](#), [concat](#), [contains](#), [map](#), [equals](#), [less-than](#), [more-than](#), [if](#), [and](#), [or](#), [hash](#), [avg](#), [split-list](#), [for-each](#), [Not](#), [IsEmpty](#), [date](#), [pad](#), [rating-average](#) and [rating-sum](#), [article-info](#), [attribute](#), [tracker-field](#), [category-present](#) but
 - we need a way to do such advanced calculations across field items (like columns in a spreadsheet) and with filters and conditions. Something like a "compute" param added to [PluginList](#), [PluginListExecute](#) and [PluginCustomSearch](#)? (It should be able to add any data to the calculation. Other fields in current tracker but also other data elsewhere in Tiki)
 - calculation across tracker items and across trackers
for ex. a tracker of payments and sum all payments, all payments of a specific period or a specific user or group
calculate certain values of one tracker with values of another tracker based on relations, for example similar to object-links, or dynamic-object-list ites.
- [Spreadsheet](#) can do some calculations (but the integration in Tiki should be improved)
 - Spreadsheets are too brittle for when data is collaborated on (It becomes important to lock cells) or constantly appended to (or formula needs to be carefully planned)
 - We want a database input form (and in some cases, data importing) and not humans adding more line to a spreadsheet
 - Spreadsheet are not great at data versioning (and auditing)
- [PluginPivotTable](#) is great to select part of the data, and to do some calculations
- [TableSorter](#) is very nifty and can do some calculations:
<https://mottie.github.io/tablesorter/docs/example-widget-math.html>
- [PluginFittesse](#) is very cool and should be extended to all future development. All projects with calculations should use such tests to make sure of data integrity as their formulae evolve.

Why is it important to calculate values server-side?

For Spreadsheet, PivotTable, TableSorter and WickedGrid, calculations are done client-side (by the user's browser) and thus, we can't use formulae the same way we use [Mathematical Calculation Tracker Field](#) calculations. This would be needed for

- Unattended operations. Ex.: trigger actions via [PluginListExecute](#) to send alerts (based on calculations) or reports (with results of calculations)
- For security reasons: imagine we are using these formulae to manage complex tax scenarios in the shopping cart. We don't want a clever user to [Greasemonkey](#) the Tiki [Shopping Cart](#) into giving them a special price. Same thing for form validation.
 - *It could be checked against server side value if the special price is valid but I understand your concern*

Questions

- Should we be looking into a way to do Server-side JavaScript?
 - Either JS via PHP: [Server-side JavaScript](#)
These are just ways of coding in js instead of coding in php. There is nothing here which is easier to do in js than in php except if you haven't learnt php
 - Or via [PhantomJS](#) which will now be easy to deploy via [Composer Web Installer](#)
 - Yes, why not? ☐
Isn't PhantomJS the opposite of what we want to achieve? PhantomJS is for automated regression testing of web-based software (very nice). It is an emulation of what a user can do. If we double check what the user does while he is doing it using PhantomJS, I believe it will mess with the user's session and actions, like doubling the orders, etc

Ideas

- Should [Advanced Ratings](#) be added to [PluginList](#)?
- Should we have a new concept of value which can be set? Perhaps [PluginValue](#) which can be set, updated, emptied, used, etc.
 - Like taking [Dynamic Variables](#) to the next level
 - Like [PluginAttributes](#)
- Could be useful for [Stats](#) (e.g. how many times page was edited, most "time consuming" page, calculate man-hours, etc.) and Action Reports?

Opinions

- I like spreadsheets because it's easy to visualize the calculations. I'd like diverse values all over Tiki to be in the spreadsheet, and from here, to use the spreadsheet formulae and then, for these resulting values to make their way into wiki pages, reports, alerts, etc. and in some cases into other spreadsheets ☐
- I prefer trackers over spreadsheets, cause trackers collect data dynamically. There is an option to import data from trackers to spreadsheets in Tiki, Although this seems to be a one-off import (not dynamically updated). Further more the input to trackers is made by pre-defined forms, which cannot be altered by most users (user groups) and work even for Anonymous in some Use Cases, where spreadsheets are difficult to maintain, specifically to lock partially. Where in spreadsheets users can alter data from other users and mix up fields, this can easily be prevented with Trackers. Imho the biggest disadvantage of trackers is the current lack of cross-item and cross-tracker calculation, which same time seems to be the biggest advantage of spreadsheets - calculation across columns and sheets.

Use Cases

See [Calculations Use Cases](#)

Code and libraries

- <https://github.com/markrogoyski/math-php/>
- <https://hoa-project.net/En/Literature/Hack/Math.html>
- <https://packagist.org/packages/clue/graph>
- <https://packagist.org/packages/brick/math>
- <https://packagist.org/packages/lstrojny/hmmm-math>
- <https://packagist.org/packages/zendframework/zend-math>
- <https://github.com/NumPHP/NumPHP/>

Related links

- [Visual Programming](#) from these calculations, we can want certain actions
- [Mail Merge](#): because the text could vary depending on the result of a calculation
- [Tracker Rules](#)
- [alarm field type in trackers](#)
- [Task/action markup for meeting notes and plans \(like Twiki Action Tracker Plugin\)](#) -> Could be a task based on a condition
- [PHP 5 Math Functions \(W3Schools\)](#)
- <http://php.net/manual/en/ref.math.php>
- [Be careful about floating point number precision](#)
- [Merge Tiki Spreadsheet into Tiki Trackers](#)

alias

[Calculation](#)