# SEARCH USABILITY (AUTO-COMPLETE): ENSURE GOOD LIVE SEARCH PERFORMANCE BY USING A MINIMUM AMOUNT OF CHARACTERS AND A MORE RELAXED SEARCH ACTIVATION TIMEOUT DELAY TO AVOIDING HAMMERING YOUR SERVERS

Status
🟢 Open

Priority
4 Good thing

Task
Search usability (auto-complete): Ensure good live search performance by using a minimum amount of characters and a more relaxed search activation timeout delay to avoiding hammering your servers

Add a timesheet
Add a timesheet for this task

Description
This is an idea of a potential performance optimization.

Even with a fairly fast manticore-based search, Tiki should avoid creating excess work/stress (roundtrips/latency/IO/etc.) on the servers when doing live searches (i.e. filtering search as the user

types). This will be useful for:

- The global searchbar
- Any field autocompletion that needs to search through extensive data (ex: things that need to search through the whole database, not just "categories names" for example)
- Automatic related items and duplicates detection (future feature) when creating new items
- Probably makes it easier for anyone with slow/high-latency internet connections, and mobile usage scenarios.

The logic and philosophy can be seen here:

- https://github.com/getting-things-gnome/gtg/issues/281 (fixed, and proven to have made a huge difference)
- https://github.com/Drive4ik/simple-tab-groups/issues/794 (fixed, and proven to have made a huge difference)
- https://gitlab.gnome.org/GNOME/nautilus/-/issues/1731 (a good general argument and theoretical overview of why this is relevant)
- https://gitlab.gnome.org/GNOME/gtk/-/issues/4133#note_1486168 (empirical proof provided)
- https://gitlab.gnome.org/GNOME/gedit/-/issues/398

It seems Tiki 24 already does have *some* (?) search typing timeout search activation delay built into it (if the top-right searchbar's behavior is any indication), however it has two problems:

- It's probably too quick/agressive, it feels like 100 or 200 miliseconds or something. I would recommend something like 500ms "after the user has stopped typing". Or if you want to get extra fancy, you could have a logarithmic/degressive timeout value based on the number of characters; ex: wait 1.5 seconds if there is only 1 or 2 characters typed, 1 second if there is 3 or 4 characters, 750ms if there is 5, and 500ms if there is more than 5 characters in the search query (no need to go lower than 500 ms afterwards).
- It currently searches even if there are very few characters. If you type a single character and wait a fraction of a second, it will run a search, and that will be very slow. In a global search, it should never immediately search for something with only 1 or 2 characters, and for anything less than 4-5 characters it should wait a fairly long time before triggering the search, as a search is more expensive when the query is imprecise and the search domain is large.
- If it's 0 characters, it should probably issue a "cancel / clear / reset search" command or something (assuming searches can be cancelled).

https://dev.tiki.org/item7938-Elasticsearch-vs-MySQL-vs-MariaDB-Search-usability-Timeout-based-search-activation-algorithm

Files

1. [2023 05 10 181800](#) 🔍

Emails

Assigned to
developer

Assigned by
adrien.maloba

Keep informed
benoit.gregoire, jonny.bradley, victor.emanouilov, xavier.depedro

Also visible to
1- Visible to Anonymous (Public content)

Related

- 🟢 5 Nice to have: Search usability: when searching for related items to associate to a tiki tracker, search results should have a popover tooltip for details (such as extended description) assigned to developer
- 🟢 4 Good thing: Enforce duplicates detection at item creation when typing a wiki page or tracker item/task, based on the title/summary assigned to developer
- 🟢 4 Good thing: Provide a unified link insertion button and syntax, instead of distinguishing between Internal (WikiLink) vs external hyperlink toolbar action buttons assigned to developer
- 🟢 5 Nice to have: Search usability: not easy to run a titles-only search within wiki pages only or tasks only assigned to developer

Created
Tuesday 09 of August, 2022 16:18:01 EDT
by Jeff Fortin T.

LastModif
Wednesday 10 of May, 2023 18:18:43 EDT

# Comments

## User mentions
[Marc Laporte](#) 2022-08-09 17:19

[https://doc.tiki.org/User-Mentions](https://doc.tiki.org/User-Mentions) should have this

# Autocomplete services should use the search index
[Jonny Bradley](#) 2022-08-10 09:44

I think at the moment most of the autocomplete "services" use the database directly, which is bad (e.g. user autocomplete on tiki.org).

Most of them use `tiki-ajax_services.php` from line 61 onwards and is very old. They really should be moved to use new or existing controller (API) actions in `lib/core/Services`, so i would suggest moving and modernising them one by one - the calls for them should all go through `$.tiki` defined in `lib/jquery_tiki/tiki-jquery.js:971`

HTH

# Manticore
[Marc Laporte](#) 2022-08-21 22:03

[https://play.manticoresearch.com/simpleautocomplete/](https://play.manticoresearch.com/simpleautocomplete/)

# More from Manticore
[Marc Laporte](#) 2022-09-09 16:23

[https://manticoresearch.com/blog/simple-autocomplete-with-manti core/](https://manticoresearch.com/blog/simple-autocomplete-with-manticore/)

# Related discussion
[Marc Laporte](#) 2023-01-17 09:49

Federated Task-Tracking with Live Data
[https://nlnet.nl/project/FederatedTaskTracking/](https://nlnet.nl/project/FederatedTaskTracking/)

[https://m-ld.org/](https://m-ld.org/)

# Related pages on Wikipedia

[Marc Laporte](#) 2023-05-10 18:19

https://en.wikipedia.org/wiki/Typeahead
https://en.wikipedia.org/wiki/Incremental_search
https://en.wikipedia.org/wiki/Autocomplete

# We have an example on avan.tech for tasks

[Marc Laporte](#) 2023-05-10 18:22

You can test here: Add a task

# Looking on GitHub

[Marc Laporte](#) 2023-05-10 18:32

https://github.com/search?q=auto-complete+language%3AJavaScript&type=Repositories&ref=advsearch&l=&l=JavaScript&s=stars&o=desc

# More from Jeff: Use a more reasonable SearchEntry timeout activation delay to avoid spamming search backends and the UI frontend

[Marc Laporte](#) 2024-02-22 00:30

https://gitlab.gnome.org/GNOME/gnome-calendar/-/issues/1170

Search results show items that do not match the completed search query (intermediate / leftover results from previous searches)
https://gitlab.gnome.org/GNOME/gnome-calendar/-/issues/989