TOC revamp

Maketoc has some design challenges: some bugs will never be fixed. See: Maketoc

Proposal: replace by a client-side Table of Content generation using JavaScript, removing or not the current server-side maketoc

It should use https://getbootstrap.com/docs/4.0/components/scrollspy/ like Auto TOC but with

- global params which can be overridden per page (like maketoc does now.) or perspective
  - active vs inactive by default
  - Subsection titles:
    - Static (classic behavior) or dynamic (TOC sections expand and collapse depending on the viewport's location, as can be seen when scrolling on this page)
    - If static, the maximum level to display (ex.: 2,3 like in maketoc)
  - position: right, left, center, where plugin is on page

Should TOC-s be available for objects other than wiki pages?

- Another use case: long tracker forms in popups (nice to have)
  - Are you talking about forms using wiki pages as templates (pretty trackers)?

Should the presumably much less used TOC plugin (for structures) not be renamed to something like STRUCTURETOC, or STRUCTURECONTENTS?

Issues with automatic tables of contents

- When the toc is higher than the page's view-port, there should be a scroll bar.
  - It is already possible to scroll using the browser's scrollbar.
    - ok, so maybe it's been solved? This only was visible on huge tocs
      - I don't think that's what I meant, but looking at https://getbootstrap.com/docs/4.0/layout/grid/ there is a TOC scrollbar. Unlike in https://getbootstrap.com/docs/3.3/components/. But unfortunately it's not Bootstrap 4 per se which adds such a scrollbar since it is still lacking in Tiki 19.
  - Is this to enable scrolling in the TOC using the keyboard or to allow exploring the TOC without losing the current position in the document? Should this be requested upstream against Bootstrap's affix?
    - Yes, perhaps an upstream thing. Maybe solved, if not, maybe in Bootstrap 4
      - But what advantages do you see in this scrollbar?
  - This is probably solved by **r69553** and **r69554** (for Tiki 19.2).
- Table of content (TOC) - maketoc or autotoc - polluted ? (if still current)

Related discussion about the interest of keeping traditional TOC-s:

- What would be the interest of keeping server-side as a possibility?
  - Good question! Since PluginMaketoc doesn't use the regular wiki plugin code, the regular plugin helper doesn't let us show all the options. But here are the docs: PluginMaketoc and it seems to me all doable client-side. One issue though is when JavaScript is not available, like in some printing context, so it could require the same solution as for other similar cases: Converting dynamic JavaScript content into static assets. But, in any case, mPDF has another way of dealing with this: https://mpdf.github.io/what-else-can-i-do/table-of-contents.html All in all, I would get rid of maketoc (ex.: last version would be Tiki18) assuming AutoToc (or whatever it's called) can do an acceptable job.

        - maketoc does not accept any parameter other than those documented. So, you are saying that:
            1. Currently, maketoc is more configurable thanks to these parameters.
            2. Once #1 is solved, the only interest of keeping server-side would be JavaScript. Is there a context other than PDF exportation which could be affected? And if not, do you agree that once #1 is solved a client-side parameter would be useless?
    - Well, a lot of people find it currently useful. Just because the server-side plugin has a bunch of *wontfix* bugs does not guarantee that the client-side plugin won't have other, different unfixable issues. In this context I understand *make it optional* as let people switch to the new version when *they* decide it better fits their needs.
        - But do you see any such issue?

Option A: maketoc with a new param client-side

- *Marc* wrote:

    > *Since PluginMaketoc doesn't use the regular wiki plugin code*

    .
    I'm strongly in favor of consistency and code reduction. If there is some plugin standard or regular tiki/wiki code we should be sure we go to that direction for any plugin we improve. I'm not in favor of exception and "aliens" implementation because a project or two need it.
    I'm strongly in favor of improving Tiki as users see it more than improving code only devs see. I move from *undecided* to *reject* because I'd prefer keeping maketoc working until we get autotoc in shape.
    - What do you mean. Why would adding this parameter stop maketoc from working?

maketoc with a new param client-side

Accept	Undecided	Reject
0	2	2
	• luci • Torsten	• Jyhem • MarcLaporte

Option B: replace with a new autotoc plugin

Add a plugin which would have parameters allowing to set options of the Bootstrap TOC. There could be extra options to define how (or if) the TOC displays when the page is printed, accessed from a mobile device, or exported as a PDF file.

Having a plugin (rather than wiki page properties) would allow using TOC-s with more object types.

The location of the plugin call in the syntax could define the position of the TOC, but not necessarily (there could be an option to override that).

What would happen with the current "Automatic table of contents" tiki-editpage field? If it is removed, what would happen with the values of the field at the time of removal? And if it is not removed, what would happen if the autotoc plugin is used in a page with "Automatic table of contents" set to No?

- Marc says we should keep a field so that TOC-s can be disabled if they are enabled by default. In that case, should the field not become just a "Display TOC" checkbox?

Why would the plugin be called "autotoc"? "automatic" in the sense that the author does not have to manually copy-paste (let's say) each title? If so, is that not obvious? There is already something called "Auto TOC".

new autotoc plugin

Accept	Undecided	Reject
2	0	1
• Jyhem • MarcLaporte		• luci

- MakeToc, in its current design is fundamentally flawed in today's context (it likely was best way to do at the time it was done). It needs to understand the logic of any other plugin on the page. Even with a lot of work, it will never be reliable, which is why we need a revamp or a replacement. Seems to be best to start with a fresh code base.

Optional : Remove maketoc

finally remove maketoc

Accept	Undecided	Reject
1	0	2
• MarcLaporte		• Jyhem • Torsten

- Autotoc is nice, although autotoc in no means is a full or at least sufficient replacement for maketoc. — Torsten Fabricius
- I think we should remove in 2 or 3 years, once AutoTOC is a suitable replacement. After an LTS so sites can stay with that another 5 years, and they have ample time to transition. We could add a PluginAlias as we've done for other deprecated plugins in the past but this is not a perfect solution as parameters won't neatly match up.

What would happen with existing calls to maketoc if it is removed?
It will be just text, like if you put {thenameofapluginthatdoesnotexit}

Option C: Make maketoc generate a TOC client-side (rather than server-side)

replace maketoc code so users get new version on upgrade
Ideally, params would continue working

Accept	Undecided	Reject
0	2	2
	• luci • Torsten	• Jyhem • Marc

New options we want

- param to put toc where the plugin is vs right aligned vs...

About using mPDF's ToCs

- item6611-Allow-replacing-hardcoded-call-to-maketoc-in-page-with-the-one-produced-by-mpdf

Alias

Maketoc revamp