No Javascript

This page is to discuss future steps regarding the use of JavaScript in Tiki and report ways how Tiki can work better in contexts where JavaScript is not activated.

Related: Accessibility, Browser Compatibility, Ajax, jQuery, Security, Mobile

JavaScript pro and con

[Show/Hide]

- The basic problem
  - Todo
- Proposal - Plan of action (2015 April 20)
- Discussion - Allowing JavaScript to be optional?
  - Thoughts on Accessibility by hartmans
    - Yes
    - No
- Rule set for coding (suggestion)
- Related

jQuery was added as an experimental feature of Tiki 3.0 and became the default in 4.0. Tiki will have more and more JavaScript usage to improve user experience. Ideally, Tiki has a graceful fallback for when JavaScript is not available. It is expected that look and feel won’t be as nice, but we should strive to make it work. To test, use NoScript

The basic problem

It is not about pro or contra JavaScript as a language and not about deciding whether Tiki should use JavaScript for functionality where JavaScript is needed.

"It is about "should Tiki require JavaScript" in a way that even for simple browsing the content, navigating through menus and simple basic editing tasks we want to require the user to have JavaScript enabled and processing in their browser. It is about if we as the Tiki community, Tiki project and Tiki software choose to take away from our consultants and site administrators the opportunity to create JavaScript-free websites. Do we really want to force consultants and projects that require basic no-script navigation and editing to either accept JS in their browsers or to simply go and find other software ... meaning as well for some of us to either abandon this type of projects and customers, or to need to split our resources between Tiki and new other software, or even to leave Tiki, cause things continue to go into directions, as some of us cannot follow when JavaScript is made mandatory and non-optional.

Furthermore, apart from the JS question, we should discuss in the community, which type of market we want to approach. Do we still want to address a wide market and do we still have the goal to offer an all-inclusive web tool for all kind of use cases and all kind of various user groups? Do we still address the professional market as trying to get a stable system with robust fallbacks for special situations? Do we focus on the professional market and do we respect, that we need to support our self-employed consultants to get work and to get companies into the project who can co-fund the development? Or is Tiki more like a
hobby project, where single coders commit stuff to their own needs and spleen and can just ignore the
needs of other community members who partially indeed do not code themselves, but commit other values
to the communities and who depend on Tiki generated revenues or who maintain NGOs’ web projects based
on Tiki?

(Gary’s thoughts:) I don’t necessarily agree with the description in the paragraph above and other parts of
this page, about types of people involved in the Tiki project and what their motives might be, or about the
feelings of markets and users, and some statements I believe go a little too far. But I think we all would like
to support as many types of users and organizations as possible. In my opinion, it’d be great if Tiki could be
used pretty much without JavaScript but, like many ideas about features and functionality, there’s the
question of limited resources and relative payoff for the efforts made.

In my opinion, if it isn’t too much trouble to add and maintain no-JS fall-backs for actions that normally use
JavaScript, this is a good “selling point” for Tiki and a practical value for the users who care about that, but
we are talking about an estimated 1.5% to 2% of Internet users. These are probably people and
organizations we want to support for philosophical reasons, but this needs to be considered in the context
of all the tasks that need to be done to keep Tiki rolling along.

- In April 2015 we are saying we have to decide if the Tiki project accepts that there are considerable
  numbers of users on the Internet who decide to browse the Internet with the no-script option active in
  their browsers and if the software Tiki shall allow those users by default to access content and basic
  editing on average (not massively customized) Tiki websites.

- We have to decide if we want to seize the opportunity for web developers, admins and web masters
to address no-script users as considerable target group of their web publications and to let them at
least contribute in a basic manner on Tiki based collaboration websites like intranets.

- We have to decide whether consultants can offer their customers no-script access to Tiki content or
  not.

- There are projects out in the Internet which address data-protection groups, hackers and hard-core
  administrators, who avoid allowing JavaScript being processed in their browsers. Browser plugins like
  for example “no-script” are quite popular. Just check the Firefox Plugin No-Script and see 2,283,941
downloads.

- Please mind, that if a user uses no-script or switches JS off in his browser, he cannot at all browse a
  Tiki website right now and he cannot even print a wiki page or find the necessary content in a menu.

Do we really want to leave those people out and just ignore them as potential users? Are we really so good,
that we do not need them? Do we really know for sure, that those people and the NGOs who deal with them
as target group are such a small irrelevant group? Are all the Occupy, Blockupy, data-protection, anti-data
retention activists, hardcore administrators, Snowden sympathizers, NSA citizens etc. irrelevant small
groups, which should not be addressed by Tiki and all in general and by our consultants in specific?

This is the question here - and again: tech-wise we are talking about robust fallbacks for navigation and
basic editing, not about advanced functionality. Do we really hide access with JS and obtrusively force users
to use it only to be able to READ our content or to EDIT A WIKIPAGE ? Is that the way we want to go?

Todo

1. Toolbar & Smiley rely on JS, so should not appear
   ○ Help should continue to be shown at bottom of page (like in 3.0)
2. Forms submitted by JS onchange event (e.g. theme switch) should always display submit button with at least "Go" when JS is off
   1. Done: theme switch shows "Switch" button when JS is off
3. profiles won't work

Proposal - Plan of action (2015 April 20)

At 20 April 2015 Gary, Amette and Torsten agreed on a plan of action and to suggest this as a concept to the community

• 1. Create a minimal set of rules that helps grasp the core concepts of non-JS, like "keep the information accessible"
  or said simpler: "identify where JavaScript is used, and of these places, which should have a fallback/alternative"
• 2. Codify that.
• 3. Implement across the board for Tiki 15 LTS

Please add your name to the plugin proposal to mention, if you agree to that plan or you are undecided or opposed.

<table>
<thead>
<tr>
<th>Accept</th>
<th>Undecided</th>
<th>Reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

• Torsten
• amette

The community will implement robust no-script fallbacks for Tiki 15 to some extend with the above sketched plan 20_April_2015

Discussion - Allowing JavaScript to be optional?

Thoughts on Accessibility by hartmans

I've often heard the claim that Javascript poses a barrier to accessibility particularly for the blind, who are likely to use text-mode browsers. It's certainly true that there are a number of vocal blind users who are
interested in getting as much as possible out of text mode browsers; it's quite common to see discussions on the blind linux users list of trying to use text-mode browsers to download Youtube videos, podcasts and the like. Of course, it's quite common to see discussions of accessing Linux machines from someone's primary DOS-based computer or migrations from DOS to Linux as a primary operating system. Which is to say that there's certainly a long tail in the blind community.

However, all the major browsers do have accessibility that is good and that works with Javascript. There is Chromevox for Chrome. Orca, Voiceover, and Narrator come as part of the OS for Linux/Gnome, OS X, and Windows. There are additional screen readers for Windows if you don't like Microsoft's offering. There are screen readers that typically come with Android and Apple has an offering for its phones and tablets. All of the above work well with Javascript.

As a blind user, I find that Javascript actually improves the efficiency with which I can use some web interfaces I use on a regular basis. As an example, especially with the new keyboard navigation, Google Play Music has a nice web interface. Openerp, an open-source ERP platform, also benefits significantly from use of AJAX that actually makes the product more accessible.

It is possible to decrease accessibility using Javascript of course. I've filed a couple of bugs here in cases where that happened. However, from my experience, Javascript is not the enemy of accessibility.

However, as I discussed above, there are blind users who use text-mode browsers. If you depend much on Javascript, you will make the product harder for those users. However, I think it's more correct to think of it as a user community who has chosen to stick with technologies they find familiar, rather than a user community who has no other options.

Yes

- accessibility
  - possible?
    - http://stackoverflow.com/q/21235312/2459703
    - http://stackoverflow.com/questions/22153388/bootstrap-no-script-fall-back-for-drop-down-button
- barrier-free websites
  - blind people (see report and discussion at item5682
- security-focused users
  - Tor Browser Bundle has NoScript installed by default
- Restrictions imposed by the owner of the client computer
  - Public computers on US Army bases
- JavaScript may be blocked by censoring governments like China (see e.g. first Edit here at Stackoverflow
- machine readable sites and search engines
- legal obligations
- Tiki philosophy: make things optional
- respect the environment
- people might deactivate it or use noscript plugin (e.g. Firefox)
  - tracking of users, third party JavaScript are reasons for this behaviour
  - desktop ressources are wasted, powerusers with many browser windows and many tabs
- NGOs don't like it
- Tiki admin access should be easy
- usage of text-based browsers, e.g. lynx
No

- hinders progress and modernization of Tiki with features that are universally popular
- modern touch screens and dropdown menus
- better functionality
- better user experience
  - some accessibility tools use JavaScript (see https://github.com/paypal/bootstrap-accessibility-plugin)
- support Ajax and Tiki services
- JavaScript is a language not a feature - unusual to make entire languages optional
- burden on code and volunteer coders - more difficult to maintain and understand code
  - Results in significant duplication of code in some cases
- cost/benefit - benefit to small user set has not yet been proven to outweigh the cost
- wisdom of “stuck-in-time” approach to JS issues is questionable and not accepted by many, including in the Tiki team
  - Other potential solutions not explored, and exploration is discouraged
  - Respect the environment which includes those who disagree with the "stuck-in-time" approach
- danger of making the most commonly used functions the most backwards - a significant overall negative for Tiki's appeal to users
  - May result since coders will be less likely to enhance or modernize with JS-based enhancements in areas where JS must be optional

Rule set for coding (suggestion)

- Restrict optionality requirement to discreet actions where considered absolutely necessary
- Optionality only to be considered for most basic features and the most basic actions within those features
- Optionality not required in following areas (not an all-inclusive list):
  - Admin areas (i.e., anything requiring tiki_p_admin_xxx or tiki_p_feature_admin permission), any feature where Ajax services are used (e.g., Trackers), Perspectives (requires jQuery UI), Spreadsheets
- Basic features where optionality should be considered:
  - Wiki, File Gallery, Article, Blog
- Basic actions where js should be optional within the basic features:
  - Edit
  - Delete

Related

- Accessibility
- TemplateStylingProposals
- WebAccessibilityInitiative
- UX+Team