



Citation for published version:

Research360 2013, *Research360: Recommendations for Improvements to Sakai Documentation..*

Publication date:

2013

Document Version

Early version, also known as pre-print

[Link to publication](#)

Publisher Rights

CC BY-ND

University of Bath

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Research360

Recommendations for Improvements to Sakai Documentation

June 2012



This work is licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Background

The Jisc-funded Research360 project at the University of Bath aimed to integrate data deposit seamlessly into the research workflow, by enabling data to be transferred from a virtual research environment directly into a data repository. The first stage of this involved developing an integration between the Sakai virtual research environment and the SWORD2 deposit specification. Development work on the integration between SWORD2 and Sakai proved to be technically far more challenging than anticipated. As such, the work was suspended in late 2012. A report detailing these difficulties is available from the Research360 project.

A frequently encountered difficulty during the work was the poor quality of Sakai's documentation. In order to contribute to improvements in this area, this document contains a list of recommendations that, if implemented, are likely to improve the experience of future developers who are new to Sakai.

Recommendations

The following list suggests some ways in which improvements could be made, ordered by how easy it is anticipated they would be to implement.

1. On the Sakai wiki, replace the search box provided by the Confluence software with a domain-restricted search box for a major search engine e.g. Google – for example, a Confluence search for “authorization” returns many results for “authoring” instead. Ensure that the Sakai wiki pages are indexed by Google and other major search engines.
2. Update documentation for the *sakai.properties* file used for localisation of a Sakai installation. There are currently two sources of information: example files (one containing commonly needed customisation, the other every possible item that can be customised), and an out of date Word document listing the customisable values with minimal information on what they mean. Not all fields have clear functions or acceptable values that can be inferred from their names. This documentation on the Sakai wiki is quite out of date.
3. Volunteers from the community should regularly scan the Sakai wiki to identify and then either fix or remove broken links.
4. Documentation matching frequently asked or anticipated questions, such as how the Sakai authorisation model works, should exist in the Sakai wiki.
5. Easy to find and regularly managed pathways into the Sakai wiki should be created and these should be linked directly from the Sakai website page for developers. These could pick up common issues for developers. Although the Programmer's Café goes some way toward this, it is not yet sufficient alone. In particular, links to the Programmers Café and other developer areas of the Sakai wiki are conspicuously absent from the Getting Started page for Technical Contributors¹.
6. Discoverability of APIs and their Javadocs should be improved, particularly for classes which are shipped as binary overlays and are therefore not in the standard source

¹ <http://www.sakaiproject.org/technical-contributors>

distribution. It is not at all clear to new Sakai developers that the Sakai CLE Kernel provides a wide array of useful APIs and extension points.

This might involve a central index of all versions of each module, along with some way of determining which module a particular class belongs to. Without detailed knowledge of the project it is not immediately obvious whether a particular class will be part of the API, the kernel, or needs to be found in the individual release Javadocs.

7. Developers should be encouraged to include more comments in their work and to add descriptions to functions, constants, etc. If correctly formatted, the latter would then automatically appear in generated Javadocs.
8. The Sakai wiki would benefit from some usability testing of access to information.
9. The Sakai wiki would benefit from both removal of out-of-date information and the addition of more up-to-date information. In addition, links from discussion documents to information about the resolution of the discussion would be particularly useful.
10. Some overarching management of the Sakai wiki content and structure might help to reduce proliferation of out-of-date pages.
11. The wiki pages on development environment set up should be altered to make clear that using an integrated development environment (IDE) such as Eclipse is optional, and not required for successful Sakai development. The high memory requirements of IDEs are particularly exposed by Sakai, which itself requires a lot of memory.