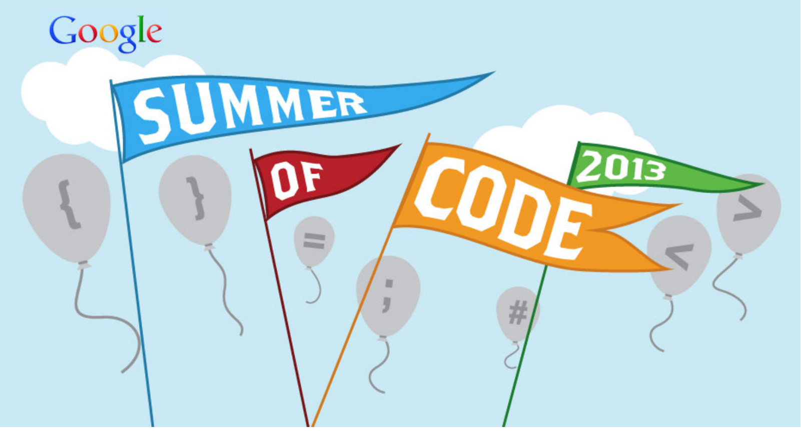
Google Summer of Code 2013 ideas list



On this page

- Overview
- About Sakai
- Community
- Technology
- Ideas

Overview

Are you interested in contributing to the Sakai project as part of Google Summer of Code 2013? *Then this page is for you!*

Scroll down for an introduction to Sakai, project ideas proposed by our mentors, and tips on making your application. Sakai has applied as a mentor organization again this year; please check the GSoC 2013 site for program schedule dates.

- Sakai's profile page and participant application form on Melange (GSoC's administration site)
- Our GSoC projects blog and Google group
- Former years' idea pages (2009, 2010, 2011, 2012), Melange homepage and Google Code site
- GSoC FAQs and timeline, also on Melange
- GSoC-related posts on Google's Open Source Blog
- Our application template

About Sakai

Sakai Collaboration and Learning Environment (CLE) is a powerful Learning Management System / Virtual Learning Environment (pick your preferred term), used by over 350 educational organisations and actively maintained and developed for. Sakai Open Academic Environment (OAE) is a first-principles redesign which is currently in active development and pilots in our community. Sakai CLE's ecosystem is very healthy and OAE is a completely new codebase, so expect the two to co-exist for some time.

Sakai CLE

The original Sakai project is very widely used every day and has over 4 million educational users. Usually deployed for a whole institution, users create 'sites' for learning management (e.g. associated with lecture courses), research collaboration (e.g. on a particular research project or research group), project collaboration (e.g. for course designers or final-year projects) or e-Portfolios (for learners to assemble portfolios of work they can use to reflect on and evidence their skills). Each site is a shared space for its members, and its creator chooses a suite of 'tools' to use within the site. Users work in one tool at a time. There are many built-in tools, e.g. file sharing, announcements, wiki, forum; and even more contrib tools, e.g. opensyllabus.

You can learn more about Sakai CLE on the official information page.

Sakai OAE

Sakai OAE embodies a complete new design philosophy. It aims to 'unsilo' the tools concept by simply making them content to be embedded in the site. It also aims to complement the outward-looking experience of learners and researchers by by being more social and more 'permeable', being part of the web as well as being part of the institution. OAE offers

- a revamped user experience, with a focus on usability
- a powerful and easy to use content authoring capability, including the ability to re-use content from elsewhere
- social networking features for students and academics
- improved teaching and learning workflows
- increased scalability and integration
- ease of use for software developers
- easy to deploy, to help people try it out
- reduced cost of ownership via low demands on institutional IT support

You can learn more about Sakai OAE: read an early proposal, or check out the official information page.

Community

Most people working on Sakai work at educational institutions using it, so in effect it is developed by universities and colleges for universities and colleges. The community is still run on open source principles, but this is a bit different from archetypal OSS projects where people contribute in their spare time, so we sometimes make a distinction by calling it "community source" rather than "open source". We use the Educational Community Licence 2.0 (FAQ), which is OSI certified and is a minor variant of Apache 2. The Sakai community organises itself and its work using a typical suite of OSS tools:

- an official project site for letting everyone know what we're doing, why they should be interested and how to get involved
- a project blog to keep everyone up to date and share what's on our minds
- mailing lists for support and discussion (OAE kernel development uses a Google group)
- a wiki (Confluence, also kindly contributed free by Atlassian) for plans and documentation
- an issue tracker (JIRA, also kindly contributed free by Atlassian) for managing bugs and feature requests
- source version control system for managing collaboratively-developed code (Subversion for Sakai CLE, git for Sakai OAE)
- an irc channel #sakai on freenode.net

Organization Administrators

- Carl Hall
- Seth Theriault

Project Mentors

- Aaron Zeckoski
- Zach Thomas

Technology

Sakai CLE

CLE is developed and a series of standard Java web applications (usually called tools) running in a Java servlet container (commonly Apache Tomcat). It uses the Spring Framework as a base for the Sakai CLE Kernel and Component Manager which manages the Sakai services and allows the Sakai tools (webapps) to communicate.

Individual tools (webapps) use a variety of technologies (e.g. Spring MVC, RSF, JSF, Velocity, etc.) but plug in to the CLE Kernel to access core services.

Ideas

This list is currently building, while we recruit mentors. Each mentor gets to propose their own project ideas.

Once you've got to grips with Sakai yourself you might find you've thought of an idea you'd love to work on. That hasn't happened very often in the past but we love it when it does. You'll still need a Sakai mentor though so you'll need to get on the Sakai-dev mailing list as early as possible, whip up some enthusiasm and recruit yourself a mentor.

---

0. Example project

Mentors: please enter your proposals here. Proposals should summarise what the project is about, provide some motivation for it, briefly outline important technologies and skills, and if necessary a short description of the kind of candidate you think the project would suit. Also, a little advice on getting started can help candidates make a good application.

Mentor: <mentor name>

Tracker URL / Discussion page: <confluence link if available>

Initial Assignment (for accepted students during socialization period, plus anyone who can't wait to get started)

Project Idea: CLE Package Manager

Mentor: Zach Thomas

A demo install of Sakai 2.9.1 weighs in at over 800Mb. This makes for long startup times, high memory demands, and a slow feedback loop for developers. The cause of this is redundancy of the included dependencies (there are 139 complete copies of sakai-kernel-util-1.3.0.jar, for just one example!) and an everything-but-the-kitchen-sink approach to the makeup of the basic system. The solution is better modularity and a good package management system. It should be possible to start with just authentication, authorization, and sessions and add everything over and above that a la carte, using package management. It should be just as easy to update Sakai as it is a smartphone.

Project Idea: CLE Stateless Application Server

Mentor: Zach Thomas

When you run a cluster of Sakai application servers, every user session "sticks" to one particular server (i.e. each of their requests comes back to the same machine). This causes problems for elasticity (being able to add and remove cluster nodes according to demand), failover, and rolling updates. The solution is not to store any state on an application server (with the exception of some caches, which can be thrown away at any time with no trouble). User session state should be kept as small as possible and then clustered with something like Redis, Riak, or memcached.

Project Idea: CLE Command-Line Project Bootstrap

Mentor: Zach Thomas

Developers wanting to add capabilities to Sakai have a very steep road ahead of them. The reason for this is the amount of infrastructure and process you have to understand to get even something simple working. The solution is tools to simplify startup that make all the uninteresting choices for you. The model for this is something like the Ruby on Rails experience: rails new superblog. A new project should be possible with some markup, some javascript, and a few simple model objects. With today's modern javascript frameworks (e.g. AngularJS) we don't need anything from the server other than some resources rendered as JSON.

Project Idea: CLE R&D, Evented Request Handling

Mentor: Zach Thomas

From the very beginning of the servlet specification, servlet containers have used a single thread per request. The problem with this approach is that it's possible for the thread to be idle while it's waiting for some time-consuming part of the request to be fulfilled (such as a web service call or an expensive database query). Furthermore, the overhead required for threads means that beyond a few hundred, performance falls through the floor. This places a significant constraint on concurrency when we want a service to scale to hundreds of thousands or even millions of users. The solution to this is a combination of a pool of threads for performing processing-intensive operations along with non-blocking threads using asynchronous callbacks. See Akka and vert.x for implementations of this approach.

Project Idea: Mail Sender Refresh

Mentor: Carl Hall, Seth Theriault

The Mail Sender is a tool for sending emails to various members of a course. It is written in RSF but would greatly benefit from being updated to something like Wicket or JSON feeds with a JavaScript driven front end. A success project would be to rewrite/refactor Mail Sender to be feature complete with the current offering and clear away bugs.

If you have questions or want to discuss your ideas, try **sakai-dev list**, then the **project mentor** directly (unless you don't have one yet). We will always try our best to respond and give advice, but please be patient as sometimes we have a lot of questions to answer.