

Mercury: The Sakai Developer's Portal

April 10, 2005

*Please direct questions or comments about this document to:
Glenn R. Golden, ggolden@umich.edu*

Mercury is a Sakai *portal*. It meets the needs of Sakai developers to access their tools in a simple and direct way while developing.

As a portal, Mercury provides a URL space for Sakai; it recognizes URLs and maps them to various capabilities. Mercury's URLs recognize 3 simple cases:

- home display
- tool access
- login

Mercury is much simpler to use for development than other Sakai portals. While the **Varuna** portal from Sakai 1 could be used, Varuna is the *Site* aware portal, so you would have to setup a Site and place your tool in order to test it out. And while the upcoming **Astro** portal for Sakai 2 could be used, Astro is the *SuperStructure* aware portal, so you would have to setup a SuperStructure and place your tool within in it to test it out.

To use Mercury, all you need to do is have a Sakai tool properly configured and running, and it will be accessible directly from the *home* display or by simple URL.

Mercury provides the necessary concepts of *tool placement* and *context*, in a simple, minimalist way, just enough to support these requirements of Sakai tools.

Software

Mercury software is in the `portal` module, in the `mercury` project. It is a single Servlet. The project produces a `.war` artifact called `mercury.war`. This allows the portal to be accessed with the URL:

`/mercury`

Home Display

Accessing the portal with no path, or with the path `"/mercury/home"`, produces the home display. This has a link to *login*, and shows information about the

- current user session
- registered tools

- registered helpers
- registered test tools
- registered sample tools
- current set of tool placements

This display acts as a quick check that your tool is properly available and registered.

Tools that are intended to show sample code and are marked with the registration category `sakai.sample` show up in the samples list; those that are intended just for testing and are marked with the `sakai.test` category show up in the test list.

Tools that are intended to be used as a helper tools and are marked with the category `sakai.helper` show up in the helper list. Helper tools are not directly accessible.

Tools listed in the main, sample and test lists can be invoked by clicking on their tool id.

Tool Invocation

To invoke a tool, Mercury must have a tool placement and a tool context, as well as the tool id. The default context used in Mercury is “mercury”. A tool placement is created as needed for each tool invocation; this is unique by the tool id and context. So each tool in the “mercury” context has a different tool placement, and the same tool in different contexts have different tool placements. Mercury remembers these placements while it is running; they are reset when the server resets.

Mercury recognizes the following URL pattern as a tool invocation:

/mercury/tool.id/context/path for tool

where:

- *tool.id* is the tool’s registered tool id, such as `sakai.chat`, and
- *context* is the context string
- *path for tool* is the path info string sent to the tool when invoked

The URLs generated in Mercury’s home page display default to use the “mercury” context, but you can enter any other context string with a direct URL.

Tools are given the full browser window in which to interact. To “return” from a tool, enter the / or /home URL to get back to the home display.

After invoking a tool, you will see the placement listed in the home display. If the placement accumulated any configuration information, this will be visible in the display.

Home Display

Accessing the portal with no path, or with the path “/mercury/home”, produces the home display. This has a link to *login*, and shows information about the

- current user session
- registered tools
- registered helpers
- registered test tools
- registered sample tools
- current set of tool placements

The Mercury home display starts by showing the current session information:

Session

```
session: df40064c-cb92-46a2-80fa-66178dd3df49 user id: 000-00-0000 enterprise id:
ggolden started: Apr 9, 2005 accessed: Apr 9, 2005 inactive after: 1800
no tool session established
```

- Tools that are intended to show sample code and are marked with the registration category `sakai.sample` show up in the samples list; those that are intended just for testing and are marked with the `sakai.test` category show up in the test list.
- Tools that are intended to be used as a helper tools and are marked with the category `sakai.helper` show up in the helper list. Helper tools are not directly accessible.
- Tools listed in the main, sample and test lists can be invoked by clicking on their tool id.

Sample Tools

These are the tools registered with Sakai, categorized as *sakai.sample*:

id	title	description	configuration
sakai.sample.tools.browser	Browser	A browser sample tool.	b=2 a=1 test=reg-value
sakai.sample.tools.acme	JSF sample	A sample JSF tool (calculator?).	
sakai.sample.tools.jsf1	JSF sample	A sample JSF tool.	

108

109

110 The tool placements that are currently being tracked are displayed. If the tool has

111 configured its placement, these values can be seen in the display.

Tool Placements

These are the placements we are tracking so far:

id	context	tool	configuration
531fd7d9-4bff-48d2-001f-665b3bb1315b	root	sakai.sample.tools.browser	a=10 test=placement-value

112

113

114

115 In a production Sakai portal, placements would be explicitly created and permanently
116 persisted. Mercury takes a short cut here to create placements automatically, and track
117 them only for the run of the server. Be careful not to get the wrong idea about tool
118 placements from this!

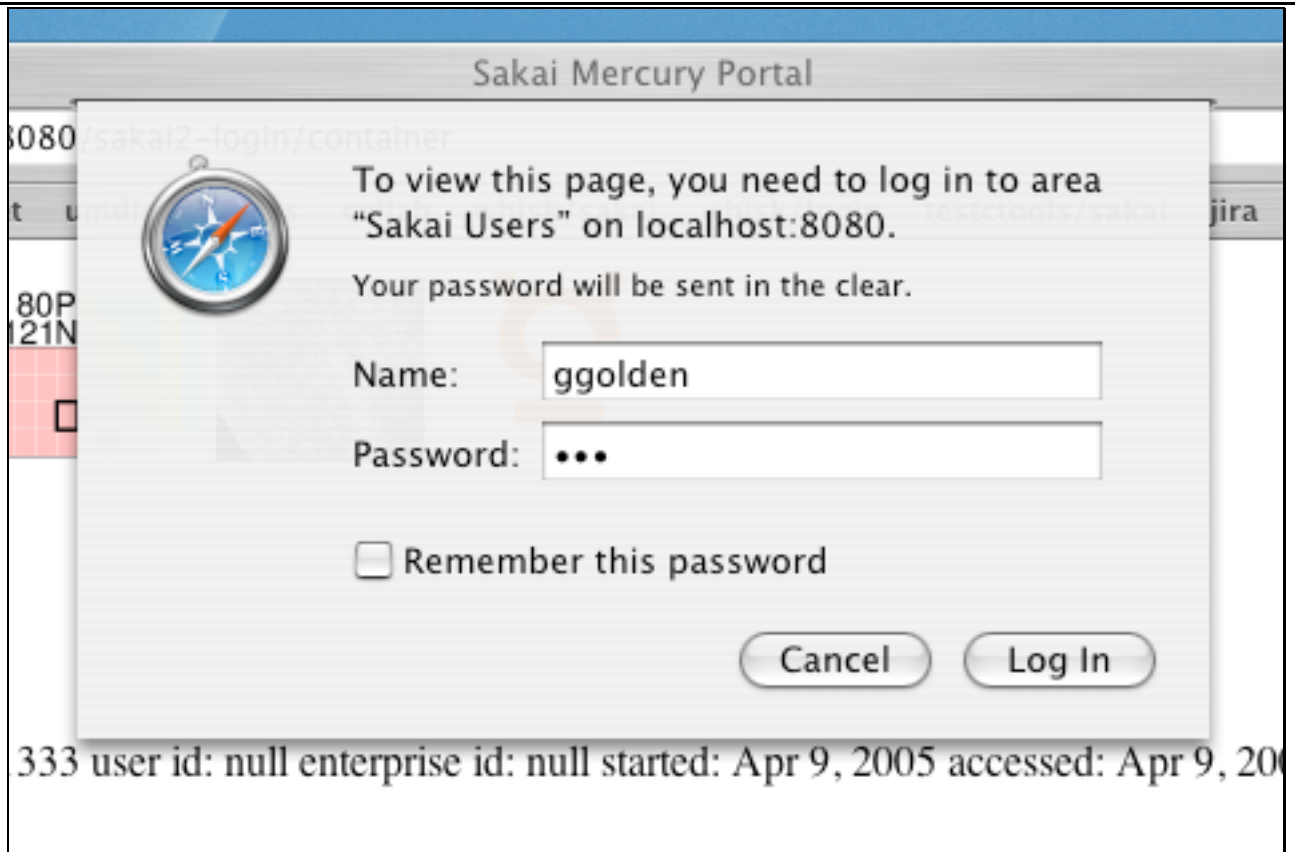
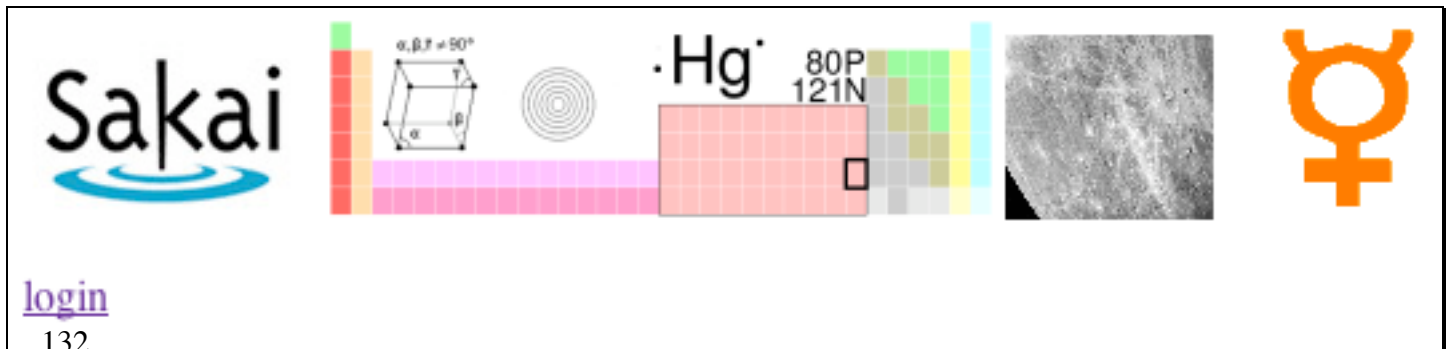
119 Login

120 The login link at the top of the home display invokes the *helper tool* `sakai.login`.

121 This login tool knows how to recognize container authentication, and otherwise how to
122 work with the Sakai Authentication API (a Sakai Common API) to authenticate the end
123 user.

124

125 If the Authentication component installed in your Sakai supports container
126 authentication, and you have setup your sakai2-login web application's web.xml and your
127 tomcat appropriately, the login will use basic http authentication to get your user
128 enterprise id and password. Once tomcat is happy with you, the authentication
129 component will lookup your end user records based on the enterprise id reported by
130 tomcat, and register your user uuid and enterprise id in the Sakai session.
131



133
134
135 You can of course hook up any other form of container authentication; this http basic
136 authentication is just an example.
137
138 Otherwise, the login process will present a dialog to collect the user enterprise id and
139 password

140

name:

password:

141

142

143 Consult other documentation for more details about login and authentication.

144