

# Home

## Contrib: Sakai Groovy Shell

### News

- **June 10, 2009:**
  - SGS2 is now available. For more details, please visit [SGS2](#)
- **May 18, 2009:**
  - An initial preview release of SGS2 is available now.
- **May 10, 2009:**
  - I have started to write a new version of SGS called SGS2. This version uses the Google Web Toolkit (GWT). The source code is available at [GitHub](#)

### Overview

- I recommend using [SGS2](#) instead of SGS.
- The Sakai Groovy Shell (SGS) projects allows you to execute Groovy code within Sakai. SGS provides two ways to do this:
  1. Telnet connection to a remote Groovy shell
  2. Sakai Groovy Shell Tool, which allows you to enter and execute groovy code

### Getting Started

- SGS has the following dependencies
  - [Contrib: sakai-wicket](#)
    - This is the dependency for the Sakai Groovy Shell Tool, which is implemented using Wicket
- Building SGS in sakai\_2-5-x
  - SGS will deploy Groovy v1.6.2 into tomcat/shared/lib/

```
svn co https://source.sakaiproject.org/svn/sakai/branches/sakai_2-5-x/  
cd sakai_2-5-x  
svn co https://source.sakaiproject.org/contrib/wicket/sakai-wicket/branches/sakai_2-5-x-wicket-  
1.3.5/ sakai-wicket  
svn co https://source.sakaiproject.org/contrib/groovy/sgs/branches/sakai_2-5-x/ sgs  
edit sakai_2-5-x/pom.xml  
- adding sakai-wicket, and sgs  
mvn clean install sakai:deploy
```

### Example connecting to the remote Groovy Shell via telnet

- To access SGS, telnet to the server at the default port (6789) where Sakai and SGS are installed.
  - e.g. telnet localhost 6789
- Note: The SGS default configuration only allows to connect from localhost. To change this, edit the SGS' component.xml file:

```
<bean id="org.sakaiproject.sgs.service.api.GroovyRemoteShellService"  
      class="org.sakaiproject.sgs.service.impl.GroovyRemoteShellServiceImpl"  
      init-method="init" destroy-method="destroy" singleton="true">  
  <property name="serviceEnabled">  
    <value>true</value>  
  </property>  
  <property name="portNumber">  
    <value>6789</value>  
  </property>  
  <property name="allowedClientIps">  
    <list>  
      <value>127.0.0.1</value>  
    </list>  
  </property>  
</bean>
```

- When the connection is established, you will see something like:

- ```
# telnet localhost 6789
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Groovy Shell (1.5.7, JVM: 1.5.0_16)
Type 'go' to execute statements; Type 'help' for more information.
groovy>
```

- Now you are ready to enter some Groovy code. The following is an example that lists all the users

- ```
groovy> import org.sakaiproject.component.cover.ComponentManager;
groovy> userDir = ComponentManager.get('org.sakaiproject.user.api.UserDirectoryService');
groovy> def userList = userDir.getUsers();
groovy> userList.each { user -> out.println user.getDisplayName() }
groovy> go
Sakai Administrator
Sakai Postmaster

===> null
```

## Example Using the Sakai Groovy Shell Tool

- Start Sakai
- Add the SGS tool to the admin workspace
- Click on the SGS tool and enter Groovy code in the text area, then click on Submit
- All the output is written to tomcat's catalina.out file
  - First it prints the Groovy source code that you have entered and submitted
  - Then it prints any output resulting from the executed Groovy code

## Development

- [Source code](#)

## Project Team

Lead(s): [Thomas Amsler](#)