CASifying Sakai with CAS 3

These instructions are for CASifying Sakai with CAS 3 and have been tested on Sakai 2.8, but should work for Sakai 2.6 and Sakai 2.7 as well.

If you are on Sakai 10+, the process has been simplified (however the instructions below still work fine), see:

Sakai 10 Read Me

If you are looking for information on CASifying Sakai with CAS 2, read my previous article.

1) Configure sakai-login-tool's web.xml

There are two blocks you need to add to the sakai-login-tool's web.xml file.

Edit: $SAKAI_SRC/login/login-tool/tool/src/webapp/WEB-INF/web.xml

First, the filter and filter-mapping blocks; add them after any others that appear in that file as below:
Of course, you need to replace the above URLs with the URLs that are relevant to your installation.

Next, add another filter-mapping block to force requests for /container through Sakai's RequestFilter. This filter must be placed close to the top of web.xml, near:
2) Modify the login-tool's pom.xml to include the casclient.jar automatically

   Edit: $SAKAI_SRC/login/login-tool/tool/pom.xml
   And add:

   <dependency>
     <groupId>org.jasig.cas.client</groupId>
     <artifactId>cas-client-core</artifactId>
     <version>3.2.0</version>
   </dependency>

3) Modify sakai.properties

   For our requirements, we need everyone to login and logout via CAS. To do this, we need to remove the username/password boxes at the top, enable the container to handle the login via CAS, and force logouts to be handled by CAS also.

   # to include the user id and password for login on the gateway site
top.login=false

   # to let the container handle login or not (set to true for single-signon type setups, false for just internal login)
container.login=true

   # Logout via our CAS landing page. Adjust this to whatever you want.
   loggedOutUrl=https://YOUR_CAS_SERVER/applogout?name=${serverName}&url=${serverUrl}/portal

3) Rebuild the login project, restart Sakai and test.

   Clicking on the "Login" link now redirects me for authentication, I provide my credentials to CAS, and I'm then redirected back to Sakai, authenticated.

Enabled local login as well as a CAS login

   It is possible to have another login link that authenticates against Sakai itself, and we use this for users that are not in our LDAP system, but need to be able to login to Sakai

   To do that, modify your sakai.properties to add the xlogin section:
# Second login link (bypasses container auth)
xlogin.enabled=true
xlogin.text=Guest Login
login.use.xlogin.to.relogin=false