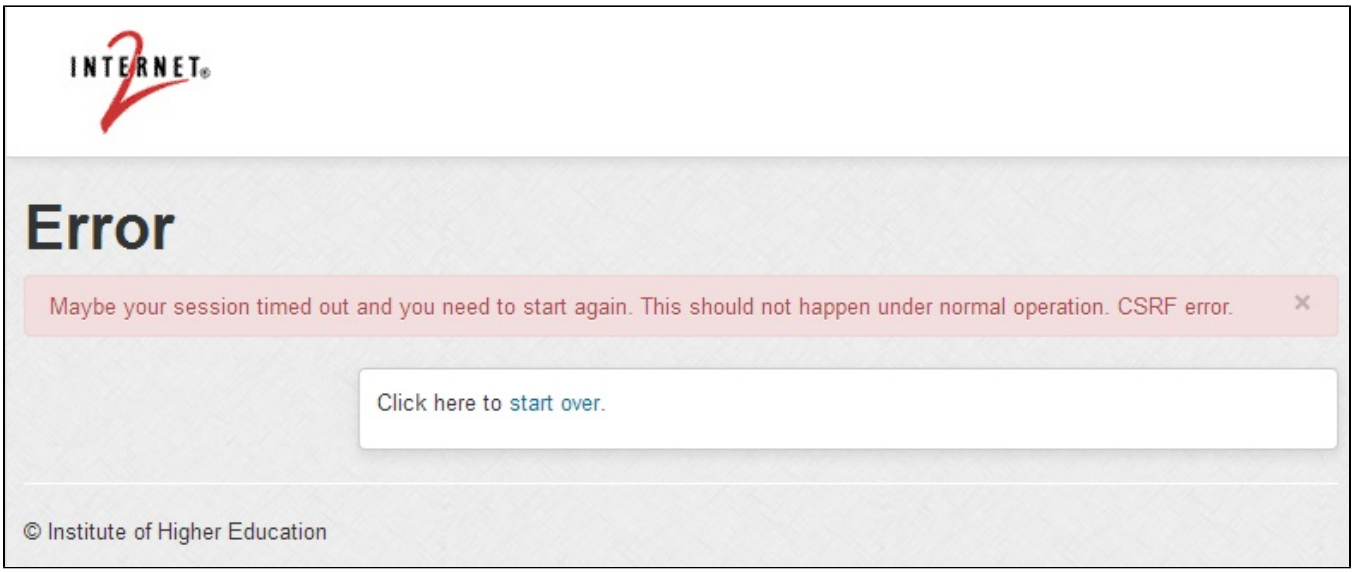
Grouper UI v2.2 error handling

There is a new servlet in Grouper used as the error page. This page will be redirected to with a code that will show an error message and let the user start over.



The URL is e.g. https://grouper.school.edu/grouper/grouperExternal/public/UiV2Public.index?operation=UiV2Public.postIndex&function=UiV2Public.error&code=csrf

This works differently that normal URLs in the Grouper v2.2 UI, the main servlet is UiV2Public.index for gets and UiV2Public.postIndex, and those are the only servlets since we cant map a wildcard in this case. So the operation is also one of those servlets, but there is a function argument which will delegate (not with reflection for security reasons) to another method. In this case it is the error routine, and the code is csrf. This code is in the externalized text file errorCode_csrf (where it is prefixed by errorCode_ and the code is the suffix)

If you are managing external users with Grouper then you need to decide on the authentication of this error page. Here is an example of an apache config that accomplishes leaving this error screen unauthenticated. If you have all of Grouper authenticated with the same authentication (i.e. not using different authentication for main grouper from external users), then just protect all of Grouper with authentication.

```
#match anything that is not grouperExternal
<LocationMatch ^/grouper_v2_0[^/]*/(?!grouperExternal/)>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user

</LocationMatch>

#match anything that is grouperExternal, but not public
<LocationMatch ^/grouper_v2_0[^/]*/grouperExternal/(?!public/)>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user

</LocationMatch>
```

If you want to protect this page but leave the public resources unprotected you can add something like this to your apache config. Test with a new session that those error pages require authentication

```
<LocationMatch ^/grouper_v2_0[^/]*/grouperExternal/grouperExternal/public/UiV2Public.postIndex>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user

</LocationMatch>
<LocationMatch ^/grouper_v2_0[^/]*/grouperExternal/grouperExternal/public/UiV2Public.index>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user

</LocationMatch>
```

If there is a 302 redirect e.g. for ajax or authentication, the browser will follow that during ajax. But it should get out of ajax and just go to that URL. The error screen will handle this with an HTTP header. So you should not get an ajax error that doesnt show the error screen

Test the error screens

1. Go directly to an error page: http://localhost:8090/grouper/grouperExternal/public/UiV2Public.index?operation=UiV2Public.postIndex&function=UiV2Public.error&code=csrf

2. Edit the Owasp.CsrfGuard.overlay.properties, temporarily uncomment: org.owasp.csrfguard.Ajax=false, go to the main app and see the CSRF error. Comment it back

3. Go to a bogus URL, should get a CSRF error: http://localhost:8090/grouper/grouperUi/app/UiV2Ma2in.index?operation=UiV2Main.indexMain

4. Unprotect the site from authentication and try to access the site

5. Setup an authenticated user who is not resolvable, login as that user (e.g. xyz)

6. Test ajax errors by going to: http://localhost:8090/grouper/grouperUi/app/UiV2Main.index?operation=UiV2Main.indexMain&throwErrorForTesting=true

dsf