# Custom Logout Feature in Dhis2 System (test in v2.20)

Edit dhis-2/dhis-web/dhis-web-commons/src/main/resources/META-INF/dhis/security.xml (near line 109)

```
<sec:http-basic />
<sec:logout logout-url="/dhis-web-commons-
security/logout_sec.action"/>
```

Edit dhis-2/dhis-web/dhis-web-commons/src/main/resources/dhis-web-commons.xml (near line 111)

```
<!--Security -->
<package name="dhis-web-commons-logout" extends="dhis-web-
commons" namespace="/dhis-web-commons-security">
  <action name="logout"
class="org.hisp.dhis.security.action.LogoutAction"
method="execute">
    <result name="success" type="redirect">/dhis-web-commons-
security/logout_sec.action</result>
  </action>
</package>

<package name="dhis-web-commons-security" extends="dhis-web-
commons" namespace="/dhis-web-commons/security">
...
</package>
```

Create new file dhis-2/dhis-web/dhis-web-commons/src/main/java/org/hisp/dhis/security/action/LogoutAction.java

```
package org.hisp.dhis.security.action;

import com.opensymphony.xwork2.Action;
import org.apache.struts2.ServletActionContext;
import org.hisp.dhis.user.CurrentUserService;
import org.springframework.beans.factory.annotation.Autowired;

/**
 * Created by nhancao on 2/26/16.
 */
public class LogoutAction implements Action
{
    @Autowired
    private CurrentUserService currentUserService;

    @Override public String execute() throws Exception
    {

        System.out.println( "LOGOUT " +
currentUserService.getCurrentUsername());
        return "success";
    }
}
```

Build: mvn clean -Dmaven.test.skip=true install

```
Result in log: LOGOUT admin
```

. . .

NOTE:

In Spring security, if you declare like this (below) mean any url pattern has match with **/abc.action** it will auto direct to spring security logout process (clear session, UserCredentials, authentication, … )

```
<sec:logout logout-url="/abc.action"/>
```

With Struts, if you declare 1 package with **namespace=”/nhancv**" like below, it mean you can access to any action had been declared in that package as **action “abc”.** So with this you can access: **/nhancv/abc.action** Once that action finish and return “**success**” message it will redirect to url: **/abc.action** and it mean url of spring security has written above.

```
<package name="default" extends="struts-default"
namespace="/nhancv">
  <action name="abc" class="TestAction" method="execute">
    <result name="success"
type="redirect">/abc.action</result>
  </action>
</package>
```

· · ·

### *Why have I to do like bridge?*

User access to **/dhis-web-commons-security/logout.action** it go to LogoutAction class, when it finished, it redirect to **/dhis-web-commons-security/logout_sec.action** of spring security to logout process. I think it is the easy way to implement and as least as possible changing code. Another solution you can just only add new **<action name=”logout” ….>** in **<package name=”dhis-web-commons-security” … namespace=”/dhis-web-commons/security">**, it mean you can access to **/dhis-web-commons/security/logout.action** to go to LogoutAction class but if you do like that you have to change all url logout for all module use logout function.

```
<action name="logout"
class="org.hisp.dhis.security.action.LogoutAction"
method="execute">
  <result name="success" type="redirect">/dhis-web-commons-
security/logout_sec.action</result>
</action>
```

Nhan Cao