

Click scope

Table of contents

1 Introduction.....	2
2 What to put in click scope.....	2
3 Using Scopes with "click" scope.....	2
4 How click scope works.....	3

1. Introduction

How many times you needed a bean that is in request scope even after a redirection? I think that you know that, if you force a redirection to another page/URL, all your request-scoped beans are erased. This is the reason that made me think about a "click" scope.

The "click" scope is like the request scope, except of the fact that it is persistent across redirection: if you put a bean in "click" scope, you will find it even after a redirection!

To use window scope see [Using Scopes with "window" scope](#)

2. What to put in click scope

You can put "click" scope almost everything you normally put in "request" scope. Beware only that, if you are relying on bean erasing in "request" scope, then probably you need a normal "request" scope.

3. Using Scopes with "click" scope

To use the "click" scope support in your web application you have to:

- extract the binary release;
- copy the "scopes.jar" file in the directory [webapp_root]/WEB-INF/lib;
- copy "lib/commons-io.jar" in [webapp_root]/WEB-INF/lib, or download the latest version from [Jakarta Commons I/O site](#).
- configure the CompleteHttpScopesFilter in front of your servlet, by modifying your "web.xml" file:

```
<filter>
  <filter-name>scopesFilter</filter-name>
  <filter-class>net.sourceforge.scopes.filters.CompleteScopesHttpFilter</filter-class>
  <init-param>
    <param-name>manager-scope-names</param-name>
    <param-value>request,click,session,application</param-value>
  </init-param>
  <init-param>
    <param-name>manager-class-names</param-name>
    <param-value>
      net.sourceforge.scopes.managers.impl.RequestScopeManager,
      net.sourceforge.scopes.click.ClickScopeManager,
      net.sourceforge.scopes.managers.impl.SessionScopeManager,
      net.sourceforge.scopes.managers.impl.ApplicationScopeManager
    </param-value>
  </init-param>
  <init-param>
    <param-name>redirect-classes</param-name>
    <param-value>net.sourceforge.scopes.click.ClickRedirectResponseRewriter</param-value>
```

```
</init-param>  
</filter>  
<filter-mapping>  
  <filter-name>scopesFilter</filter-name>  
  <servlet-name>myServletName</servlet-name>  
</filter-mapping>
```

- Installation finished! For the use see the [Quick start](#)

4. How click scope works

When the first request of "click" context happens, a new context is put in request scope. When you put something in "click" scope, you actually put it in this context object.

When a request of redirection happens, then the context object is moved temporarily in session scope, with a "request id" associated to it. The path to redirect to is modified so that it contains the request id as a cookie parameter.

After the redirection, the context object is searched in session, using the cookie parameter in the URL. After that, the context object is moved back to request. After that the cycle continues...