

This project integrates Spring MVC into OpenCms. The purpose is to provide a starting point for those who like to use Spring MVC in combination with OpenCms 7.5.0. (Although it should also work for OpenCms 7.0.5 as well).

## Versions:

OpenCms 7.5.0  
Hibernate 3.2  
Spring 2.1.0

## The project includes:

```
/lib
  /compile          <- libs only used for compiling
  /runtime          <- libs used for deployment
/nbproject          <- NetBeans project files (incl. only for convenience)
/src
  /java             <- java source, incl. rewrite filter, controller, etc.
  /jsp
    /springtest.spg
  /properties
    /opencmsspring.properties <- adjust app context and servlet name here
  /xml
    /OpenCmsSpring-data.xml
    /OpenCmsSpring-service.xml
    /OpenCmsSpring-servlet.xml
    /spring.tld
    /spring-form.tld
    /views.xml
    /web.xml
.classpath
.project           <- Eclipse projet file (included only for convenience)
build.properties  <- adjust the deployment directory of your OpenCms
install. here
build.xml
license.txt
README.TXT
```

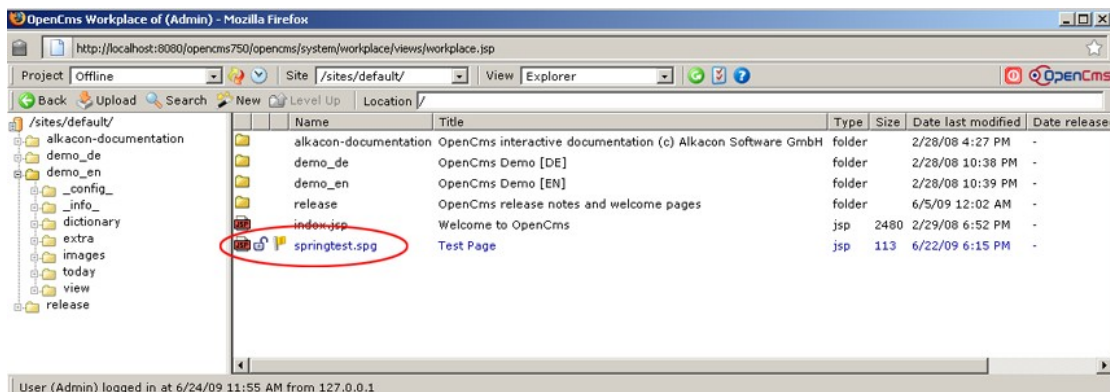
# OpenCms Spring Integration: Basic Setup

## How to use it:

1. check out the project sources from SVN:  
`http://svn.sysvision.net:9027/svn/com.sysvision.opencms.spring/`  
User: anon, Password: anon
2. adjust the deployment directory of your OpenCms application in `build.properties`
3. run: `ant deployWithXmlFiles`
  - the relevant jar file will be built at `/lib/runtime/opencmsspring.jar`
  - the built jar file and all other dependant jars (under `libs/runtime`) will automatically be deployed to your application folder (specified in the properties file)
  - the XML config files (incl. `web.xml`, spring configuration, TLD) will be copied to your application folder

Note: alternatively to running the target 'deployWithXmlFiles' you can only run 'deploy' which does the same except not copying the xml and tld files to the app dir. In some cases, this might be useful.

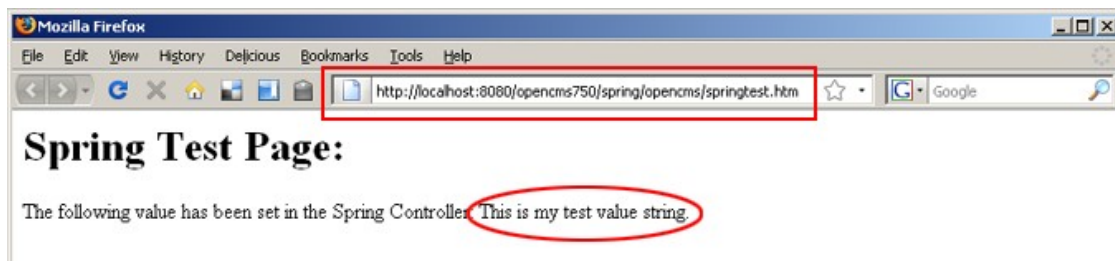
4. restart Tomcat and log into OpenCms
5. for a first test, you can place the file `springtest.spj`, which you find under `src/jsp/springtest.spj`, anywhere in your OpenCms (content type: jsp page).



The jsp contains the following code:

```
<h1>Spring Test Page:</h1>
<p>
  The following value has been set in the Spring Controller: ${testValue}
</p>
```

You will see the test value that comes from the Spring controller being displayed.



(The test value string comes from the `BasicController.java`:

```
return new ModelAndView( mvURI )
    .addObject("testValue", "This is my test value string.");
```

The basic setup is done. Now you can develop your business layer and controllers further from here.

# OpenCms Spring Integration: Basic Setup

---

## General notes / how it works:

Currently the Spring integration is working with an OpenCms setup where application name and servlet name is visible in the URL (i.e. `/opencms/opencms/` or else). Some further modifications need to be done to make it work on deployments without this URL prefix.

The name of the servlet and application context doesn't have to be `/opencms/opencms/`, it can be anything else. You can adjust the `opencmsspring.properties` accordingly.

When you want a request to be handled by Spring, you need to give the resource in OpenCms a file name ending on `*.spg`. This way OpenCms recognizes that the request should be handled by Spring first and then passed back to OpenCms.

Example: you put the sample jsp file `springtest.spg` under `http://localhost:8080/opencms/opencms/springtest.spg`

When you open the file (from the OpenCms workplace or else), the rewrite filter (`com.sysvision.opencms.spring.RewriteFilter.java`) will apply and redirect the request to

`http://localhost:8080/opencms/spring/opencms/springtest.htm`

The redirected request is then handled by Spring according to the Spring configuration (`OpenCmsSpring-servlet.xml`), using (in our sample) the `BasicController` (`com.sysvision.opencms.spring.controller.BasicController.java`).

In our sample, the `BasicController` only adds a `String` object named `testValue` to the `ModelAndView` then proceeds by displaying the view (back to `/springtest.spg`), which resides in the OpenCms VFS.

## Other useful resources regarding OpenCms/Spring integration:

OpenCms Wiki:

<http://www.opencms-wiki.org/Spring>

BearingPoint OpenCms-Modules Commons:

[http://sourceforge.net/project/showfiles.php?group\\_id=180314](http://sourceforge.net/project/showfiles.php?group_id=180314)

## Questions & suggestions:

Mathias Lin <[mathias.lin@sysvision.com](mailto:mathias.lin@sysvision.com)>

**SYSVISION Ltd.**

20/F Central Tower  
28 Queen's Road Central  
Hong Kong

Tel: +852 8199 9605

<http://www.sysvision.com>

<mailto:info@sysvision.com>

SYSVISION is an OpenCms solution provider. We provide development, consulting, training for OpenCms in Hong Kong, China, Singapore, USA and Germany.

For further information about our company visit [www.sysvision.com](http://www.sysvision.com)

24.06.09 11:45:22