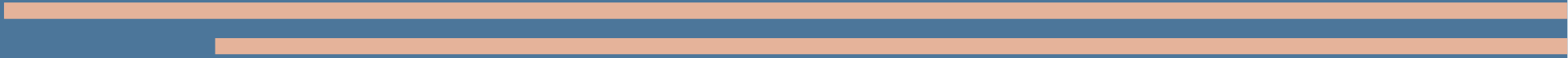




Dumping JSF

Stan Silvert
JBoss Core Developer



Overall Presentation Goal

Introduce the JSFSpy application and its WARlet architecture.



Interactive PreDemo



Introducing JSFSpy

- Provides insight into your JSF application
 - Tracks and traces HttpSession and HttpServletRequest
 - Tracks Scoped Data in each phase with complete history.
 - Rudimentary performance data
 - Requires JSF 2 and Servlet 3.0
 - Extensible via WARlets
-
-

How to install JSFSpy

- Step 1: Download the JSFSpy jar.



How to install JSFSpy

Step 1: Download the JSFSpy jar.

Step 2: Put it in your WEB-INF/lib directory.



How to install JSFSpy

Step 1: Download the JSFSpy jar.

Step 2: Put it in your WEB-INF/lib directory.

Step 3: There is no Step 3



JSFSpy Demo!



JSFSpy Data Objects

- SpyManager singleton contains all Session objects (not HttpSession, but more)
 - Session contains RequestData objects (not HttpRequest, but more)
 - RequestData contains HttpRequest, Snapshot objects (scoped data), and FacesMessages
 - All leaves are Strings
-
-

SpyManager

Session

...

Session

RequestData

...

RequestData

HttpRequest

Snapshots
(Scoped Data)

FacesMessages

All Leaves
are Strings



Creating Your Own Pages

lastLogin.xhtml:

```
<h:dataTable value="#{spymanager.sessions}" var="session">
  <h:column>
    <h:outputText value="#{session.id}"/>
  </h:column>
  <h:column>
    <h:outputText value="#{session.userId}"/>
  </h:column>
  <h:column>
    <h:outputText value="#{session.creationTime}">
      <f:convertDateTime type="both"/>
    </h:outputText>
  </h:column>
</h:dataTable>
```

Using JSFSpy with JSFUnit

```
public void testMyCustomScopeCreated() throws IOException {
    JSFSession jsfSession = new JSFSession("/foo.jsf");
    JSFClientSession client = jsfSession.getJSFClientSession();
    // Do some requests with JSFClientSession
    SpyManager spyMgr = SpyManager.getInstance();
    Session session = spyMgr.getMySession();
    for (ScopeContext scope : session.getCustomScopes()) {
        boolean foundIt = false;
        if (scope.getScopeName().equals("MyCustomScope")) {
            foundIt = true;
        }
    }
    assertTrue(foundIt);
}
```

How to install JSFSpy

- Why is it so easy?
 - How do the JSFSpy pages get into my WAR?
 - Surely JSFSpy uses filters, servlets, etc. Don't I need to declare something in web.xml?
-
-

“Thank you, Servlet 3.0”

- Servlet 3.0 says that I can put stuff in `/META-INF/resources/` and they will map to the context root.
 - All JSFSpy pages are in the JSFSpy jar under `/META-INF/resources/jsfspy`
 - I can hit this URL to get to JSFSpy:
`http://localhost/myapp/jsfspy/index.jsf`
(or whatever maps to `FacesServlet: index.faces`,
path mapping, etc)
-
-

“Thank you, Servlet 3.0”

- Servlet 3.0 also says that I can put a web-fragment.xml in /META-INF. This works just like adding stuff to the main web.xml
 - Or I can use annotations like @WebServlet, @WebFilter, and @WebListener.
 - Must use web.xml 2.5 or 3.0 deployment descriptor in WEB-INF/web.xml
 - And don't set metadata-complete=true
-
-

New Features for JSFSpy

- What would you like to see?



New Features for JSFSpy

- I don't plan to do very many.
 - I'm too lazy.
 - I'm too busy with other stuff.
 - I need help.
 - JSFSpy is open source, after all.
 - I thought the community was supposed to do the work.
-
-

Problems with getting community help

- “I'm too busy to help”
 - Developers might like the project, but not the way it is run. (Ant vs. Maven, coding standards, in-fighting, etc.)
 - Don't want to sign the contributor's agreement.
 - It takes time to get up to speed on the project.
 - Developers want to help only as long as they get to work on their own “must-have” feature.
 - Enthusiasm wanes quickly.
-
-

My Solution to Open Source Participation: WARlets

- Something I just made up a few weeks ago
 - A WARlet is a WAR add-on
 - Installed as a single jar in WEB-INF/lib
 - Requires Servlet 3.0
 - WARlet container aggregates WARlets into a single application
 - WARlets allow anyone to create their own open source WAR add-on
-
-

My Solution to Open Source Participation: WARlets

- Maybe it will be a spec someday, but there is LOTS of stuff to work out.
 - Certainly fits to make JSFSpy extensible, but it remains to be seen if the concept is generally applicable.
 - The big plus is that developers can create their own open source project for JSFSpy plugins via WARlets
-
-

WARlet Demo



Summary

- JSFSpy is a new debug/trace tool for JSF2 and Servlet 3.0
 - Collected data accessible from EL
 - Collected data also accessible from JSFUnit or other in-container code.
 - WARlets allow anyone to extend JSFSpy
-
-

For more information

- stan @ jboss.com
- <http://anonsvn.jboss.org/repos/jsfsfy/trunk/>



Questions?

