WSRP4J Configuration

Table of contents

1. Producer Configuration

1. Edit the portletentityregistry.xml file. (e.g. c:/Tomcat4.1/webapps/wsrp/WEB-INF/data/portletentityregistry.xml)
   Add a new application element for your webapp.

   The definition ID has to match the webapp name, so for the sample, this is wsrptest.

   The portlet id will map to its handle, so make up a number.

   The definition id maps to its name in the portlet.xml file like so: <webapp-name>.<portlet-name>
2. Create a portlet.xml file in your webapps/webapp-name/WEB-INF/ directory.
   Use webapps/wsrptest/WEB-INF/portlet.xml as a reference.

   The portlet name must match the second part of the definition id above.

2. Consumer Configuration (SwingConsumer)

Once you have your portlets configured on the producer side, to add a portlet to the SwingConsumer you will need to create an xml file in driver/SwingConsumer/persistence/portlets.

Use the existing files as a guide.

3. Consumer Configuration (ProxyPortlet)

You'll need some portal environment for using the ProxyPortlet component bundled with WSRP4J. The WSRP4J team has tested PlutoPortal, the minimal portal component included with the Pluto distribution.

Firstly, make sure you have your portlets correctly configured in the producer component.

1. Configure your producer endpoint urls in the proxyportlet/WEB-INF/persistence/producers/ directory. You have to create one file per producer, the naming being org.apache.wsrp4j.consumer.driver.ProducerImpl@<name>.xml, where <name> is a name you choose. If you are consuming you own local portlets, you don't have to do nothing, as there's a file org.apache.wsrp4j.consumer.driver.ProducerImpl@wsrp4j-8081.xml file with local configuration. Use one of the existing files as guidance. Be sure to assign a unique producer id to your producer. An example would be:

```
<Producer id="100">
    <markup-interface-url>
        http://localhost:8081/wsrp/wsrp4j/WSRPBaseService
    </markup-interface-url>
    <service-description-interface-url>
        http://localhost:8081/wsrp/wsrp4j/WSRPServiceDescriptionService
    </service-description-interface-url>
    <registration-interface-url>
        http://localhost:8081/wsrp/wsrp4j/WSRPRegistrationService
    </registration-interface-url>
    <portlet-management-interface-url>
        http://localhost:8081/wsrp/wsrp4j/WSRPPortletManagementService
    </portlet-management-interface-url>

    <registration-data>
        <consumer-name>WSRP4J Proxy Portlet</consumer-name>
        <consumer-agent>WSRP4J Proxy Portlet</consumer-agent>
    </registration-data>
</Producer>
```

Remember that the actual urls are configured in the tunnel.sh file in the ws-wsrp4j/tools/ directory. You could add one tunnel.sh file per producer (tunnel-ibm.sh, tunnel-bea.sh, etc) to be able to distinguish each other network traffic.

2. At this point you have to remember your producer id (e.g. "100") and your remote portlet identifier configured in the producer (e.g. "0.1").
3. Now it's time to configure PlutoPortal. Please refer to Pluto website on how to build & install it. There are two files involved, both of them in the pluto/WEB-INF/data/ directory: pageregistry.xml and portletentityregistry.xml.
4. The file portletentityregistry.xml controls the portlet applications deployed by PlutoPortal. You have to create a new application element like this:

```
<application id="5">
    <definition-id>proxyportlet</definition-id>
    <portlet id="1">
        <definition-id>proxyportlet.ProxyPortlet</definition-id>
        <preferences>
            <pref-name>wsrp_portlet_handle</pref-name>
            <pref-value>0.1</pref-value>
            <read-only>false</read-only>
        </preferences>
        <preferences>
            <pref-name>wsrp_producer_id</pref-name>
            <pref-value>100</pref-value>
            <read-only>false</read-only>
        </preferences>
    </portlet>
</application>
```

The important lines are the wsrp_portlet_handle and wsrp_producer_id preferences. The former has to be a portlet handle configured on the producer ("0.1" in step 2), the latter it's the assigned producer id ("100" in step 2). The configuration above creates a portlet

application in PlutoPortal with id "5.1", that accesses the remote portlet "0.1" in the producer with id "100".

5. Finally, the file pageregistry.xml controls the portal pages layout. You can find more information on Pluto website, but if you add a line like this:

```
<fragment name="row10" type="row">
    <fragment name="col10" type="column">
        <fragment name="p10" type="portlet">
            <property name="portlet" value="5.1"/>
        </fragment>
    </fragment>
</fragment>
```

with unique identifiers "row10", "col10" and "p10", you'll get your portlet displayed when you point your browser to PlutoPortal. The property "portlet" has value "5.1" as that is the portlet application we configured in the previous step.