Dhara User Guide

1. System Setup

1.1. Dhara Portal Setup

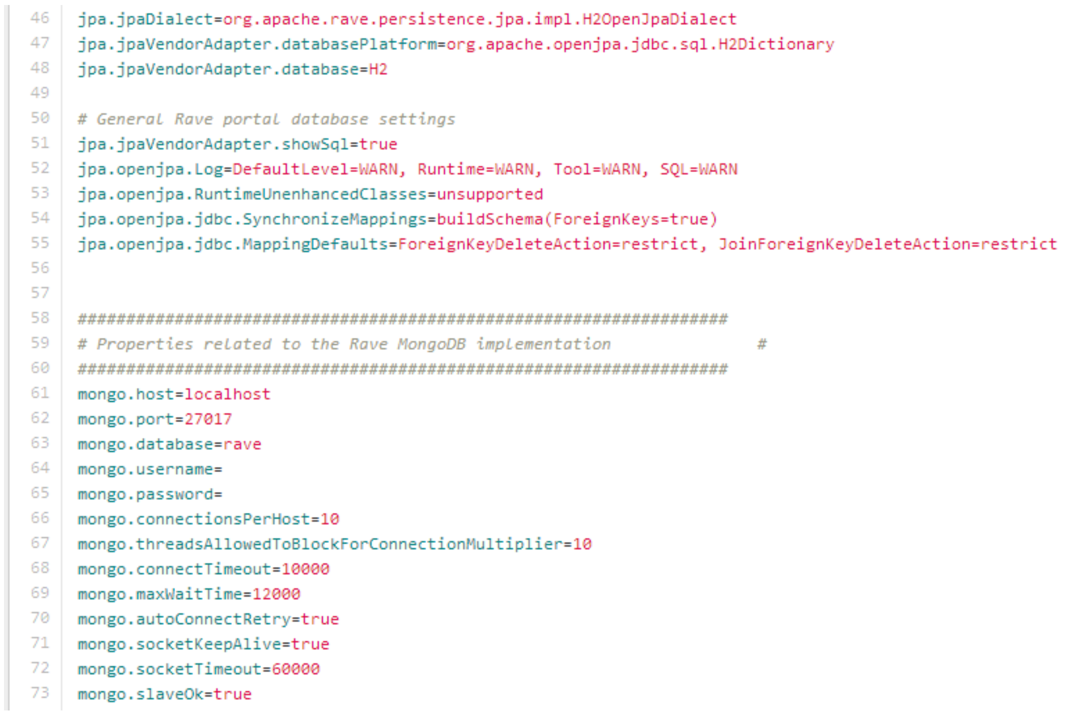
Dhara built extending the Apache Rave framework. So deployment of custom portal has some configurations to be adjusted to deploy the Dhara portal. Apache rave custom portal is contains two main projects known as *dharaportalproject-portal* and *dharaportalproject-shindig*. After checkout the portal from the Github, user needs to configure the portal settings.

There are two configuration files available in the Dhara portal. One is default property file coming from Apache Rave which is known as ***portal.properties*** and it's specifying the configurations relates to Apache rave. Other configuration file is the Dhara configuration which specifies the backend settings of the Dhara geoscience gateway setup.

- In deployment setup user first needs to configure the ***portal.properties*** file located in the "dharaportalproject/dharaportalproject-portal/src/main/resources" directory
- In portal properties configuration file, user needs to configure the Open social engine server URLs, database configurations, Wookie server URLs and mail server URL configurations
- After done with the changes user then need to configure the system setup in ***portal_configuration.xml*** which is locates in "dharaportalproject/dharaportalproject-portal/src/main/webapp/WEB-INF/conf" directory.
- In ***portal_configuration.xml*** user needs to configure the Apache Airavata server, WPS server and temporary backend server.
- Portal h2 database file is located in the db directory of Git cloned project
- User needs specify the h2 database file location in the pom file located in *dharaportalproject-portal* directory
- After all configurations are done, user needs to build the project by executing "**mvn clean install**" command inside dharaportalproject directory.
- After build completed, executing "**mvn cargo:run**" inside *the dharaportalproject-portal* directory will deploy the Dhara portal under given configurations
- To deploy the widgets of the Dhara portal, user can copy the all the widgets in the widgets folder of Git cloned project to "dharaportalproject/dharaportalproject-portal/target/tomcat6x/webapps/wookie/deploy" directory
- After completed the steps given above, user can log into the Dhara portal through default canonical user. Username: canonical and Password: canonical

Example:

portal.properties file when rave hosted in localhost



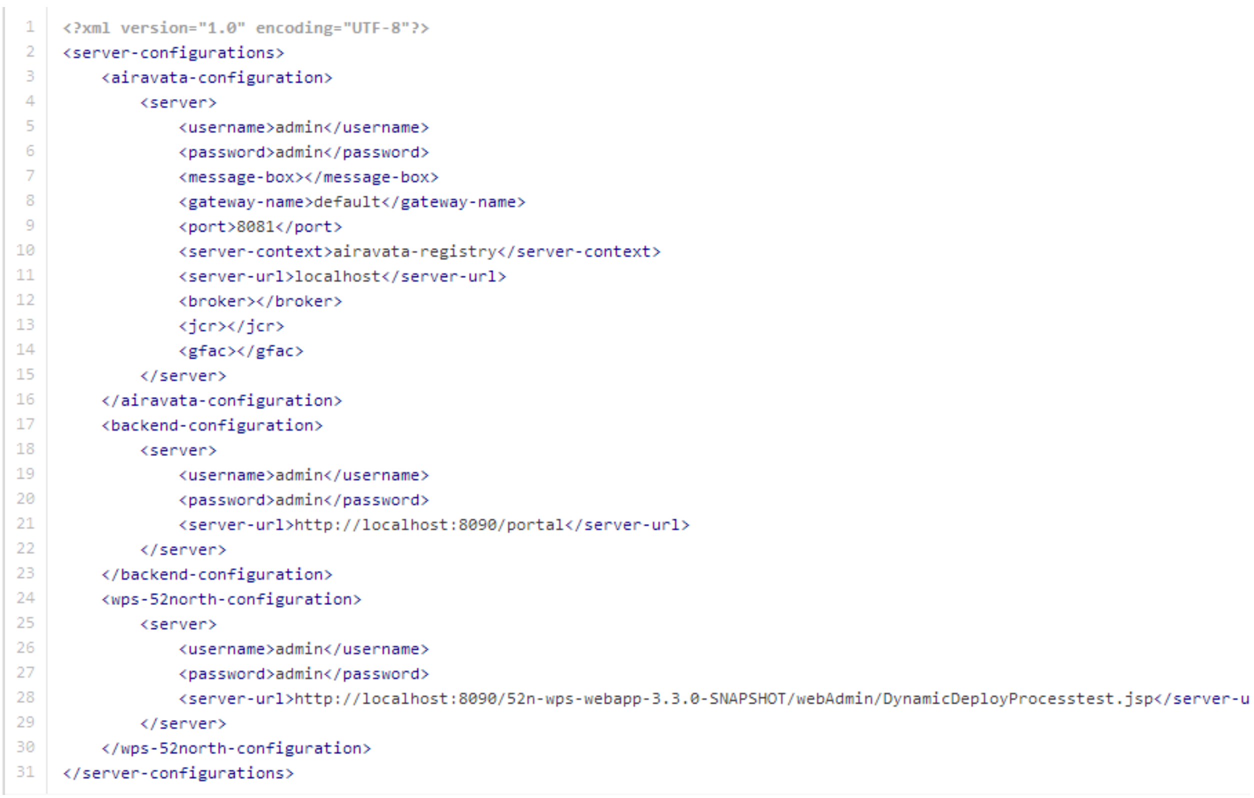
```
jpa.jpaDialect=org.apache.rave.persistence.jpa.impl.H2OpenJpaDialect
jpa.jpaVendorAdapter.databasePlatform=org.apache.openjpa.jdbc.sql.H2Dictionary
jpa.jpaVendorAdapter.database=H2

# General Rave portal database settings
jpa.jpaVendorAdapter.showSql=true
jpa.openjpa.Log=DefaultLevel=WARN, Runtime=WARN, Tool=WARN, SQL=WARN
jpa.openjpa.RuntimeUnenhancedClasses=unsupported
jpa.openjpa.jdbc.SynchronizeMappings=buildSchema(ForeignKeys=true)
jpa.openjpa.jdbc.MappingDefaults=ForeignKeyDeleteAction=restrict, JoinForeignKeyDeleteAction=restrict


###################################################################
# Properties related to the Rave MongoDB implementation               #
###################################################################
mongo.host=localhost
mongo.port=27017
mongo.database=rave
mongo.username=
mongo.password=
mongo.connectionsPerHost=10
mongo.threadsAllowedToBlockForConnectionMultiplier=10
mongo.connectTimeout=10000
mongo.maxWaitTime=12000
mongo.autoConnectRetry=true
mongo.socketKeepAlive=true
mongo.socketTimeout=60000
mongo.slaveOk=true
```

portal-configuration.xml

Portal config file for localhost deployment



```xml
<?xml version="1.0" encoding="UTF-8"?>
<server-configurations>
    <airavata-configuration>
        <server>
            <username>admin</username>
            <password>admin</password>
            <message-box></message-box>
            <gateway-name>default</gateway-name>
            <port>8081</port>
            <server-context>airavata-registry</server-context>
            <server-url>localhost</server-url>
            <broker></broker>
            <jcr></jcr>
            <gfac></gfac>
        </server>
    </airavata-configuration>
    <backend-configuration>
        <server>
            <username>admin</username>
            <password>admin</password>
            <server-url>http://localhost:8090/portal</server-url>
        </server>
    </backend-configuration>
    <wps-52north-configuration>
        <server>
            <username>admin</username>
            <password>admin</password>
            <server-url>http://localhost:8090/52n-wps-webapp-3.3.0-SNAPSHOT/webAdmin/DynamicDeployProcesstest.jsp</server-u
        </server>
    </wps-52north-configuration>
</server-configurations>
```

1.2. Apache Airavata Setup

- Apache Airavata configuration is very easy task for user
- User can have the standalone version of Apache Airavata server distribution
- After downloading the Apache Airavata server, user needs to specify the IP address of server in airavata.properties configuration file in bin directory of Apache Airavata distribution
- Then running the airavata-server.sh executable script will up the Apache Airavata server

1.3. 52^{o} North WPS Setup

- User first needs to build the WPS from the source
- WPS source uses default port of geoserver as 8094, user can change the geoserver port to a different one through wps configuration as needed
- Then deploy war distribution as a webapp in tomcat container will be done the the deployment of 52^{o} North WPS instance

1.4. Geoserver setup

- User needs to get the war distribution or the standalone version of geoserver
- Running war distribution will be done the deployment of geoserver

1.5. Backend web app setup (temporary)

- User needs specify the backend Airavata Server configuration file locates in the resource directory of the backend web application
- Then user needs to build the portal source by executing “***mvn clean install***”

- Then user can find the war distribution of backend web application in target directory
- Then deploy war distribution as a webapp in tomcat container will be done the deployment of backend web application

2. Access System Functions

The features provided through the portal are categorized into two sections based on the user role. There are set of users that are allowed only for users with admin privileges and the rest are available for all the users.

According to default configurations:

Admin user: username – canonical, password – canonical
General user: username -john.doe, password – john.doe

2.1. Access Admin user functions

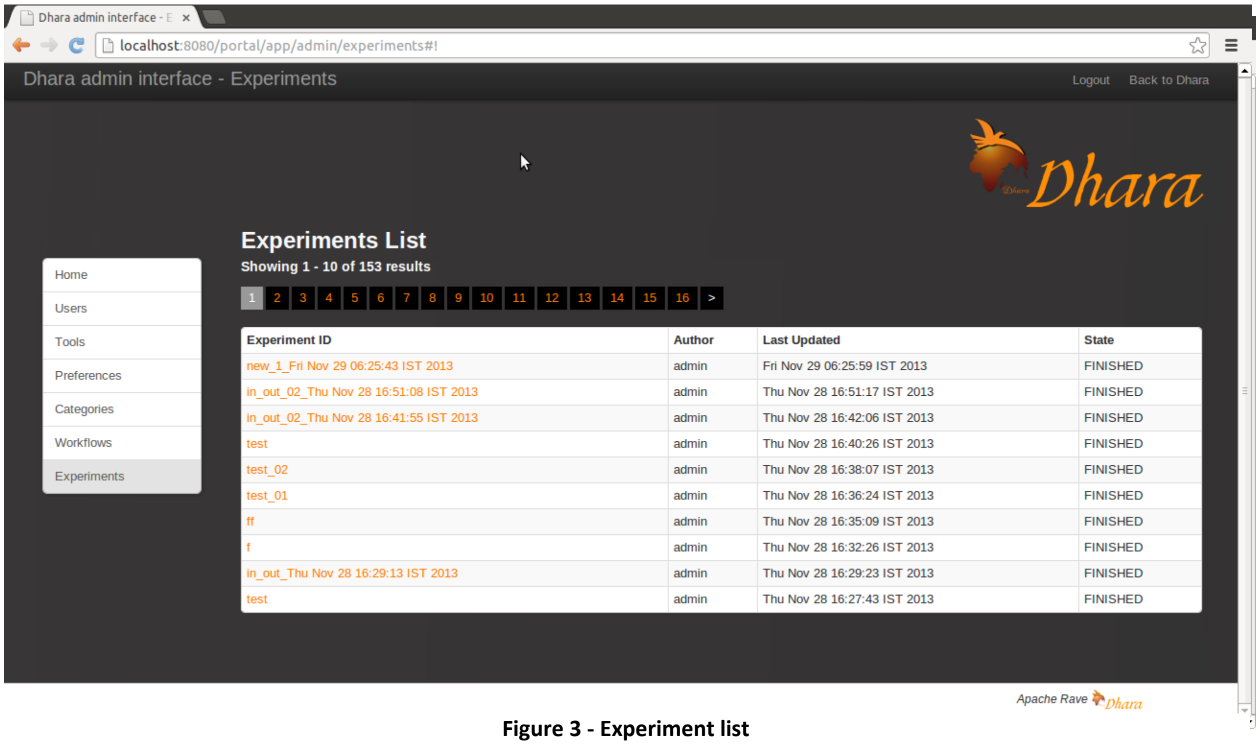
Dhara portal provides an admin user with different tools to manage and test their experiments via portal. Admin user interface can be viewed by login as an admin user and navigating to "Admin user interface".

2.1.1. Experiments

This component allows an admin user to view all the experiments he has tested via the Apache Airavata server and view other properties of an experiment such as experiment status, author of the experiment and other experiment details associated with each node. The experiment details of each node includes input values that the user has given when performing the experiment, the outputs from each functional node and the name of each node along with their service identifier.

Viewing experiment details:

- Login as an admin user and click on the "**Admin user interface**" in the top menu bar. This will navigate the user to the home page. (Figure 1)
- Click on "**Experiments**" menu item in the left side menu list. This will load all the experiments that user has tested. (Figure -2)
- Click on any experiment name that you want to view further experiment details. This will load a complete list of experiment details in a popup table. (Figure - 3)


Dhara admin interface - E
localhost:8080/portal/app/admin/experiments#!
Dhara admin interface - Experiments
Logout
Back to Dhara
Dhara
Home
Users
Tools
Preferences
Categories
Workflows
Experiments
Apache Rave Dhara

Experiments List

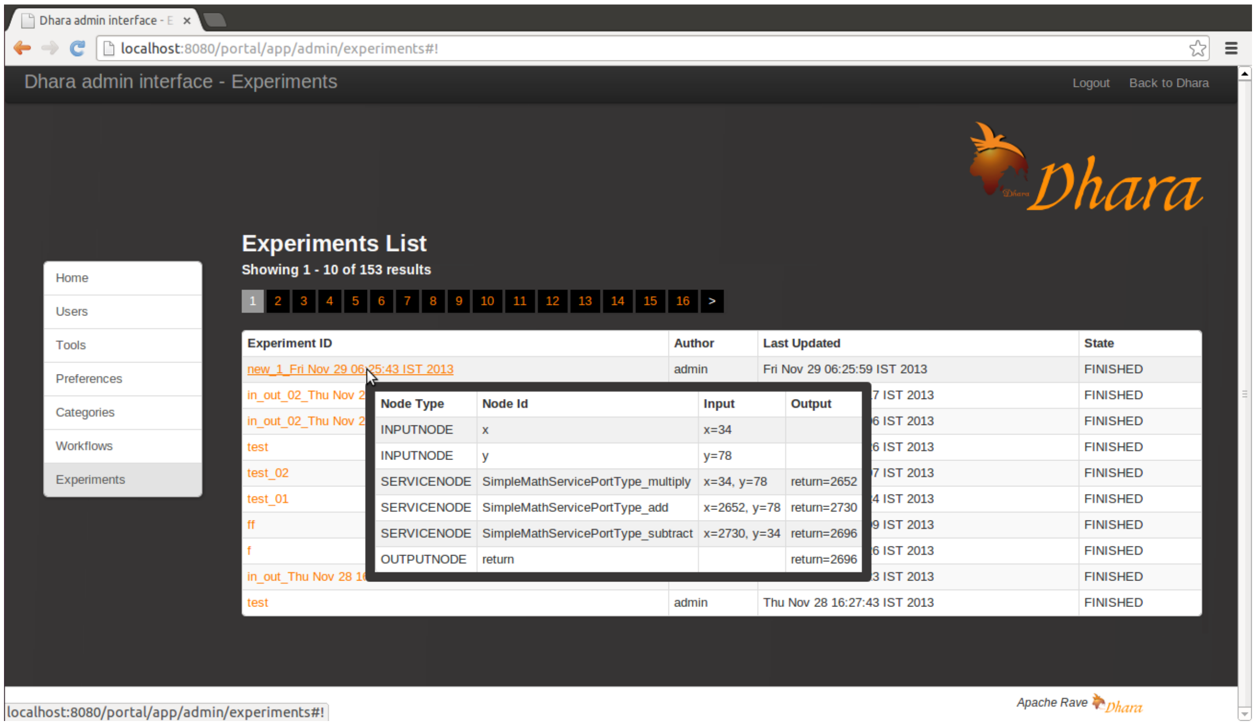
Showing 1 - 10 of 153 results

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 >

Experiment ID	Author	Last Updated	State
new_1_Fri Nov 29 06:25:43 IST 2013	admin	Fri Nov 29 06:25:59 IST 2013	FINISHED
in_out_02_Thu Nov 28 16:51:08 IST 2013	admin	Thu Nov 28 16:51:17 IST 2013	FINISHED
in_out_02_Thu Nov 28 16:41:55 IST 2013	admin	Thu Nov 28 16:42:06 IST 2013	FINISHED
test	admin	Thu Nov 28 16:40:26 IST 2013	FINISHED
test_02	admin	Thu Nov 28 16:38:07 IST 2013	FINISHED
test_01	admin	Thu Nov 28 16:36:24 IST 2013	FINISHED
ff	admin	Thu Nov 28 16:35:09 IST 2013	FINISHED
f	admin	Thu Nov 28 16:32:26 IST 2013	FINISHED
in_out_Thu Nov 28 16:29:13 IST 2013	admin	Thu Nov 28 16:29:23 IST 2013	FINISHED
test	admin	Thu Nov 28 16:27:43 IST 2013	FINISHED

Figure 3 - Experiment list

Figure 1 - Navigate to experiments


Dhara admin interface - E
localhost:8080/portal/app/admin/experiments#!
Dhara admin interface - Experiments
Logout
Back to Dhara
Dhara
Home
Users
Tools
Preferences
Categories
Workflows
Experiments
Apache Rave Dhara
localhost:8080/portal/app/admin/experiments#!

Experiments List

Showing 1 - 10 of 153 results

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 >

Experiment ID	Author	Last Updated	State
new_1_Fri Nov 29 06:25:43 IST 2013	admin	Fri Nov 29 06:25:59 IST 2013	FINISHED
in_out_02_Thu Nov 2		7 IST 2013	FINISHED
in_out_02_Thu Nov 2		6 IST 2013	FINISHED
test		6 IST 2013	FINISHED
test_02		7 IST 2013	FINISHED
test_01		4 IST 2013	FINISHED
ff		9 IST 2013	FINISHED
f		6 IST 2013	FINISHED
in_out_Thu Nov 28 16		3 IST 2013	FINISHED
test	admin	Thu Nov 28 16:27:43 IST 2013	FINISHED

Node Type	Node Id	Input	Output
INPUTNODE	x	x=34	
INPUTNODE	y	y=78	
SERVICENODE	SimpleMathServicePortType_multiply	x=34, y=78	return=2652
SERVICENODE	SimpleMathServicePortType_add	x=2652, y=78	return=2730
SERVICENODE	SimpleMathServicePortType_subtract	x=2730, y=34	return=2696
OUTPUTNODE	return		return=2696

Figure 2 - View Experiment details

2.1.2. Workflows

This feature allows an admin user to provide details of workflows that the user has created and then perform experiments from any selected workflow. Once a user has selected a workflow from the list, user can view the set inputs and their data types needed to perform an experiment executing the selected workflow. User then can provide set of inputs and execute the workflow, where user can monitor the workflow execution at runtime.

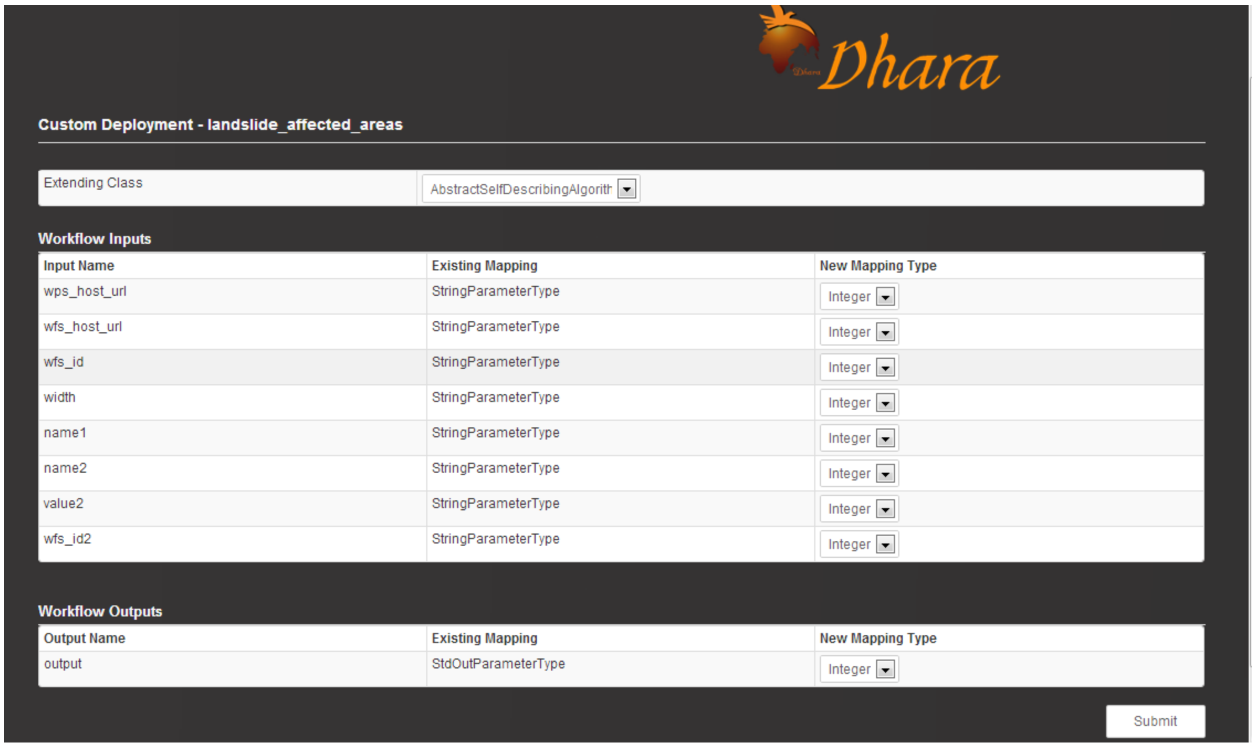
2.1.2.1. View workflow details

- Login as an admin user and click on the "**Admin user interface**" in the up menu. This will navigate the user to the home page. (Figure 4)
- Click on "**Workflows**" menu item in the left side menu list. This will load all the workflows that user has created. (Figure -5)

I. View workflow deployment functions

- In workflow list page, there are two deployment options listed for admin user. Default deployment option will directly deploy selected workflow in a 52^{o} North WPS instance by mapping workflow into WPS template.

- Custom deployment view enables user to select a user specified data type binding to generate WPS template class to deploy in 52^{o} North WPS instance. Figure 4 shows sample custom deployment view for a workflow.



Custom Deployment - landslide_affected_areas

Extending Class	AbstractSelfDescribingAlgorith

Workflow Inputs

Input Name	Existing Mapping	New Mapping Type
wps_host_url	StringParameterType	Integer
wfs_host_url	StringParameterType	Integer
wfs_id	StringParameterType	Integer
width	StringParameterType	Integer
name1	StringParameterType	Integer
name2	StringParameterType	Integer
value2	StringParameterType	Integer
wfs_id2	StringParameterType	Integer

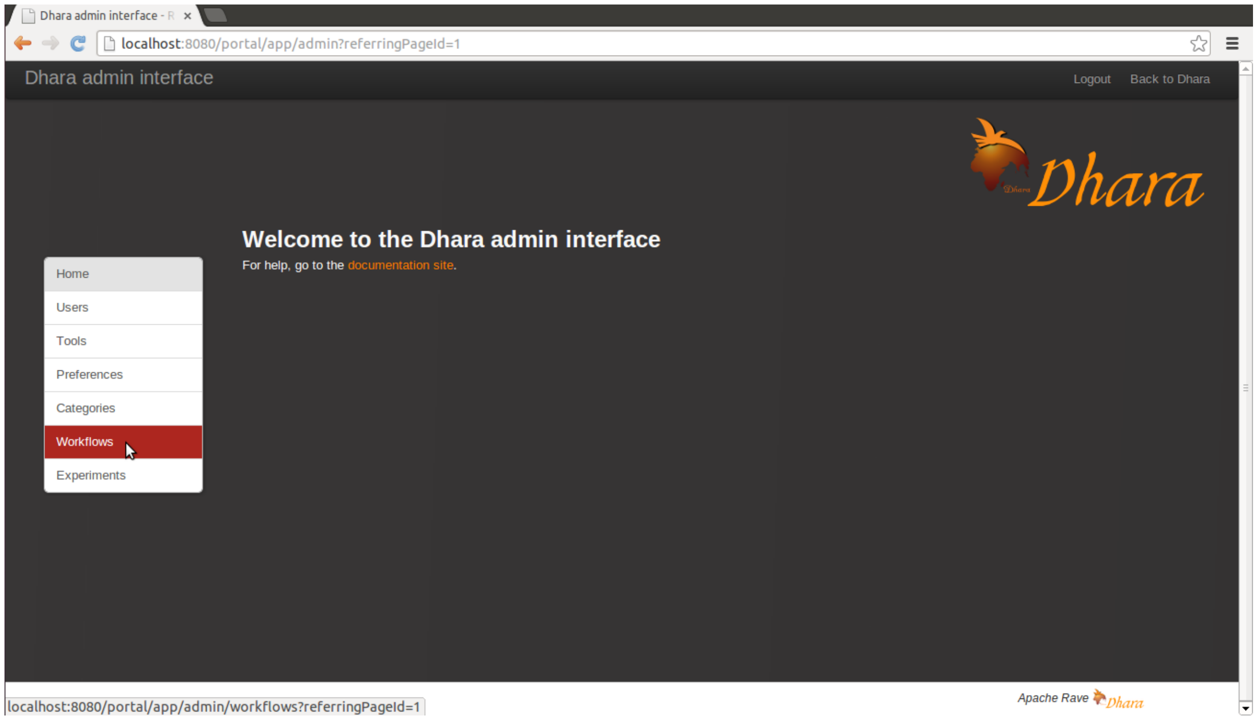
Workflow Outputs

Output Name	Existing Mapping	New Mapping Type
output	StdOutParameterType	Integer

Figure 4: Custom Deployment View

II. View workflow execution details

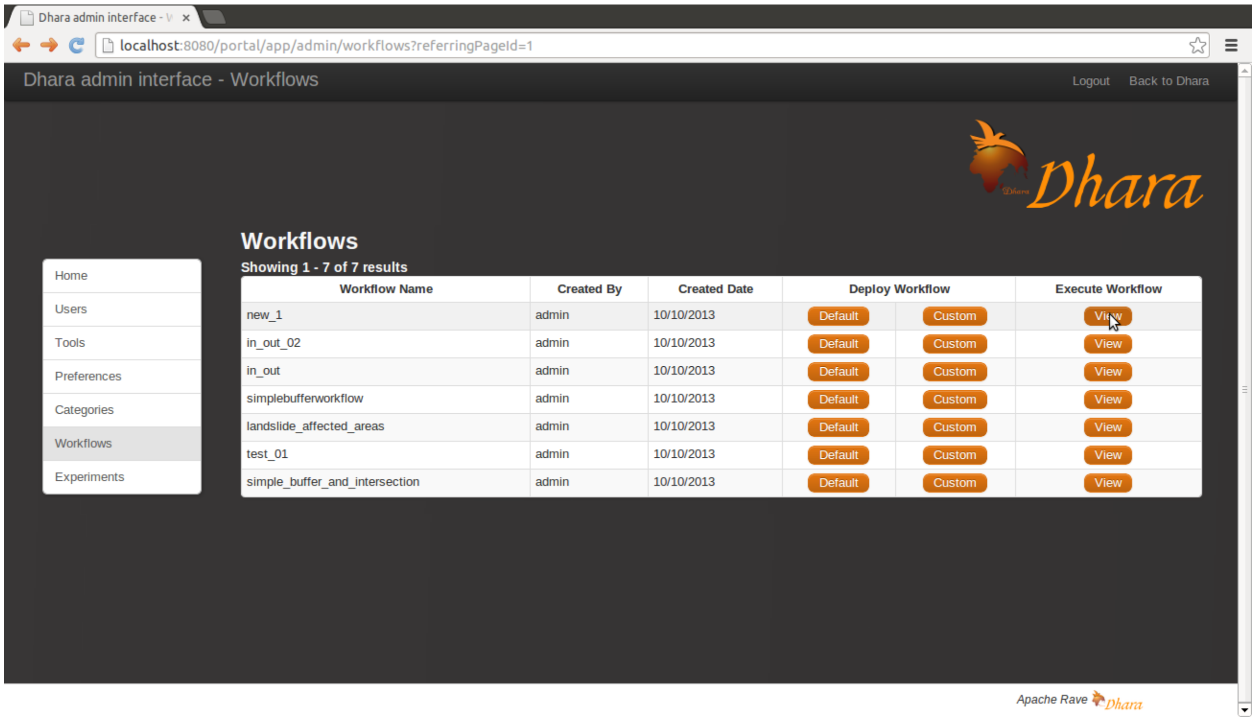
- Click on “**View**” button to view workflow description and input types.
- Enter corresponding inputs and click on “**Execute**” button to execute the workflow. (Figure -6)
- This will navigate the user to a new page which then will display monitoring details progressively. (Figure -7)


Dhara admin interface - R
localhost:8080/portal/app/admin?referringPageId=1
Dhara admin interface
Logout
Back to Dhara
Dhara
Home
Users
Tools
Preferences
Categories
Workflows
Experiments
Apache Rave
localhost:8080/portal/app/admin/workflows?referringPageId=1

Welcome to the Dhara admin interface

For help, go to the documentation site.

Figure 5 - Navigate to workflows

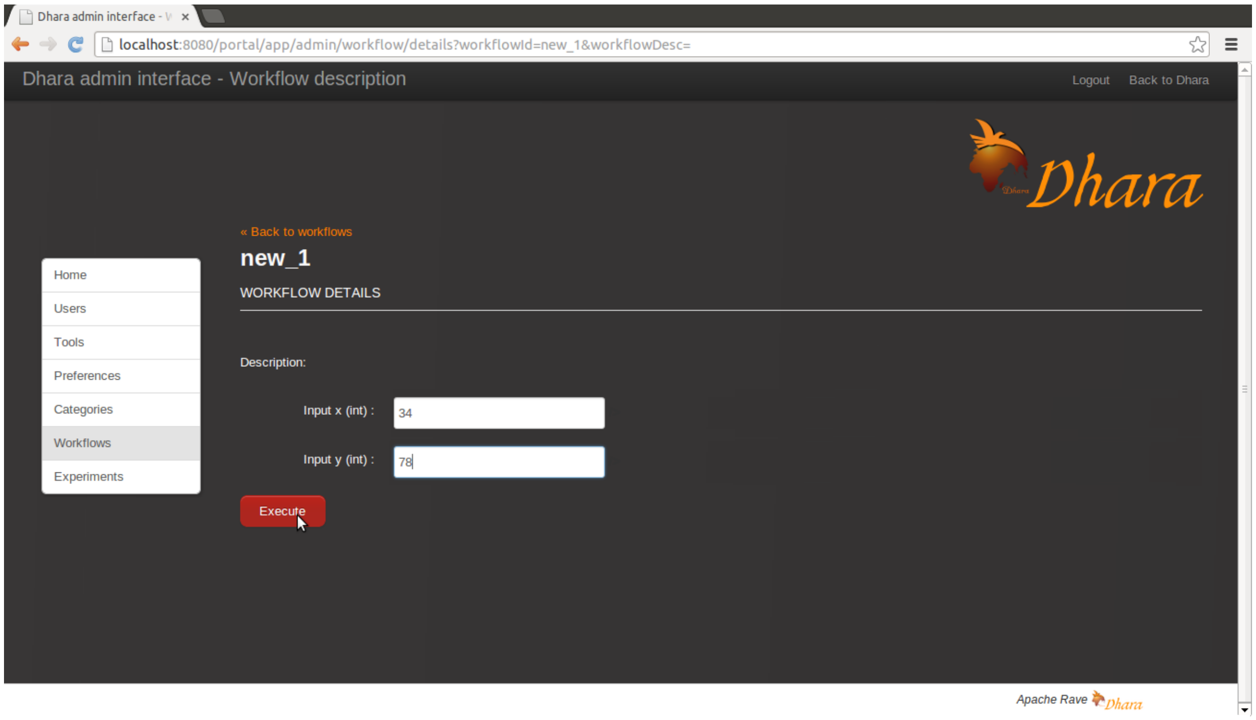

Dhara admin interface - W
localhost:8080/portal/app/admin/workflows?referringPageId=1
Dhara admin interface - Workflows
Logout
Back to Dhara
Dhara
Home
Users
Tools
Preferences
Categories
Workflows
Experiments
Apache Rave
Dhara

Workflows

Showing 1 - 7 of 7 results

Workflow Name	Created By	Created Date	Deploy Workflow		Execute Workflow
new_1	admin	10/10/2013	Default	Custom	View
in_out_02	admin	10/10/2013	Default	Custom	View
in_out	admin	10/10/2013	Default	Custom	View
simplebufferworkflow	admin	10/10/2013	Default	Custom	View
landslide_affected_areas	admin	10/10/2013	Default	Custom	View
test_01	admin	10/10/2013	Default	Custom	View
simple_buffer_and_intersection	admin	10/10/2013	Default	Custom	View

Figure 7 - Workflow list


Dhara admin interface - W
localhost:8080/portal/app/admin/workflow/details?workflowId=new_1&workflowDesc=
Dhara admin interface - Workflow description
Logout
Back to Dhara
Dhara
Home
Users
Tools
Preferences
Categories
Workflows
Experiments
« Back to workflows
Execute
Apache Rave
Dhara

new_1

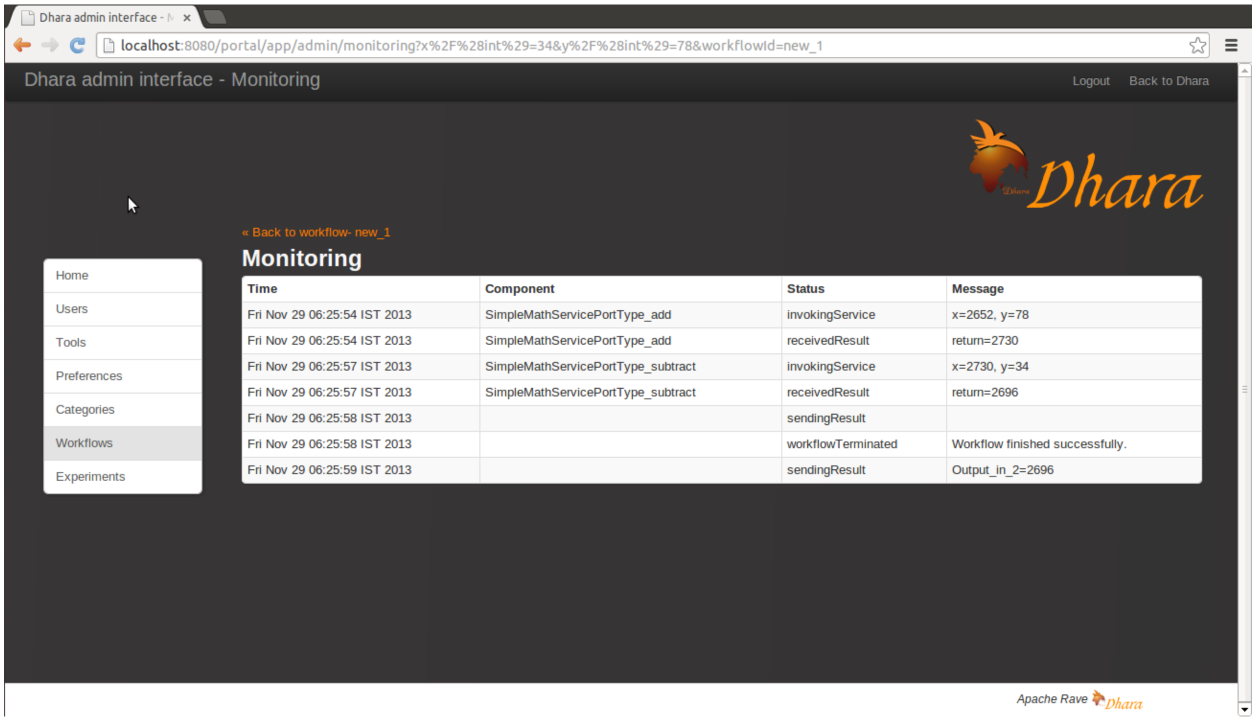
WORKFLOW DETAILS

Description:

Input x (int) : 34

Input y (int) : 78

Figure 6 - View workflow details



Dhara admin interface - Monitoring
localhost:8080/portal/app/admin/monitoring?x%2F%28int%29=34&y%2F%28int%29=78&workflowId=new_1
Logout
Back to Dhara
Dhara
« Back to workflow- new_1
Home
Users
Tools
Preferences
Categories
Workflows
Experiments
Apache Rave

Monitoring

Time	Component	Status	Message
Fri Nov 29 06:25:54 IST 2013	SimpleMathServicePortType_add	invokingService	x=2652, y=78
Fri Nov 29 06:25:54 IST 2013	SimpleMathServicePortType_add	receivedResult	return=2730
Fri Nov 29 06:25:57 IST 2013	SimpleMathServicePortType_subtract	invokingService	x=2730, y=34
Fri Nov 29 06:25:57 IST 2013	SimpleMathServicePortType_subtract	receivedResult	return=2696
Fri Nov 29 06:25:58 IST 2013		sendingResult	
Fri Nov 29 06:25:58 IST 2013		workflowTerminated	Workflow finished successfully.
Fri Nov 29 06:25:59 IST 2013		sendingResult	Output_in_2=2696

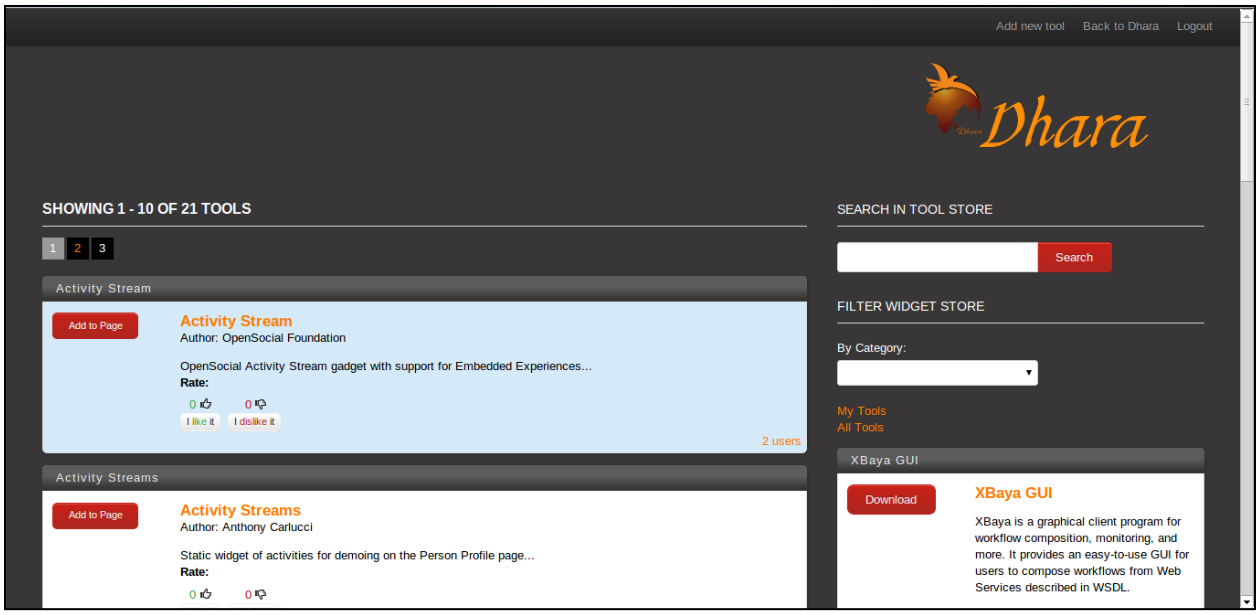
Figure 8 - Monitoring details

2.2. Access General user functions

The main functionalities available for the general user are through various tools. By default two tools are added to the Home tab in the portal: The map tool and the WPS tool. Additional tools can be added via the tool store. The new tools can be added to the existing tab or a new tab.

2.2.1. Using the Tool Store

1. Go to ‘Tool Store’ using the ‘**Tool Store’** link on the top Menu Bar.
2. Select the tool to be added to the page by clicking on ‘**Add to Page’** button of the particular tool. A description is displayed about each tool. (Figure-9)
3. We have included several tools other than Dhara – tools. In order to view only Dhara – tools, select '**Dhara**' in the category selection list.



SHOWING 1 - 10 OF 21 TOOLS

Activity Stream

Activity Stream
Author: OpenSocial Foundation

OpenSocial Activity Stream gadget with support for Embedded Experiences…
Rate:
0 0
2 users

Activity Streams

Activity Streams
Author: Anthony Carlucci

Static widget of activities for demoing on the Person Profile page…
Rate:
0 0

SEARCH IN TOOL STORE

FILTER WIDGET STORE

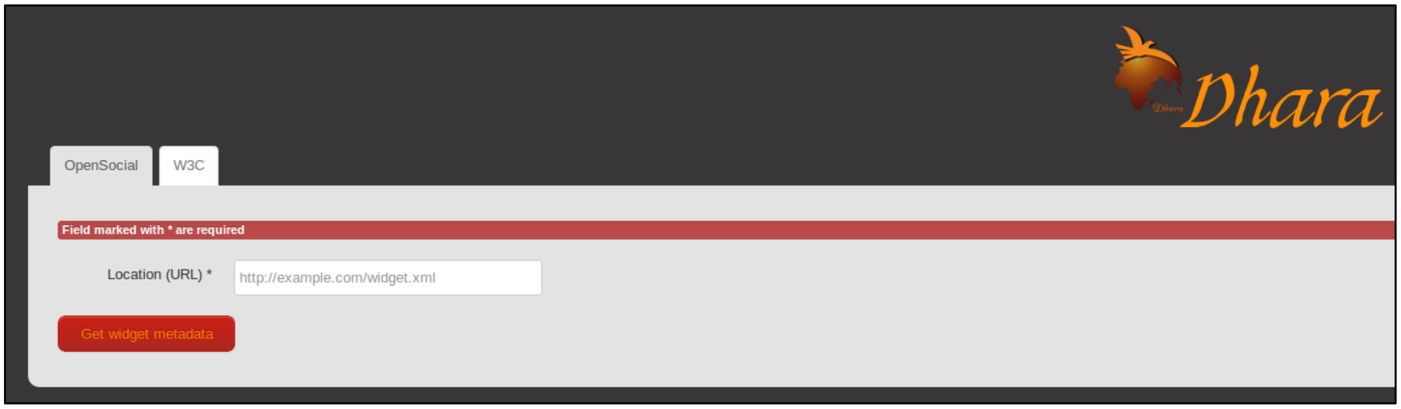
By Category:

XBaya GUI

XBaya GUI
XBaya is a graphical client program for workflow composition, monitoring, and more. It provides an easy-to-use GUI for users to compose workflows from Web Services described in WSDL.

Figure 9: Tool store

4. In order to add tools additional to the ones available in the tool store, Click 'Add New Tool' on the top Menu Bar.
5. New OpenSocial widgets can be can be imported providing a URL to an OpenSocial widget. (Figure-10)
6. New W3C widgets can be imported by browsing through the W3C widget repository provided. (Figure-11)



Field marked with * are required

Location (URL) * http://example.com/widget.xml

Figure 10: Adding a new OpenSocial gadget

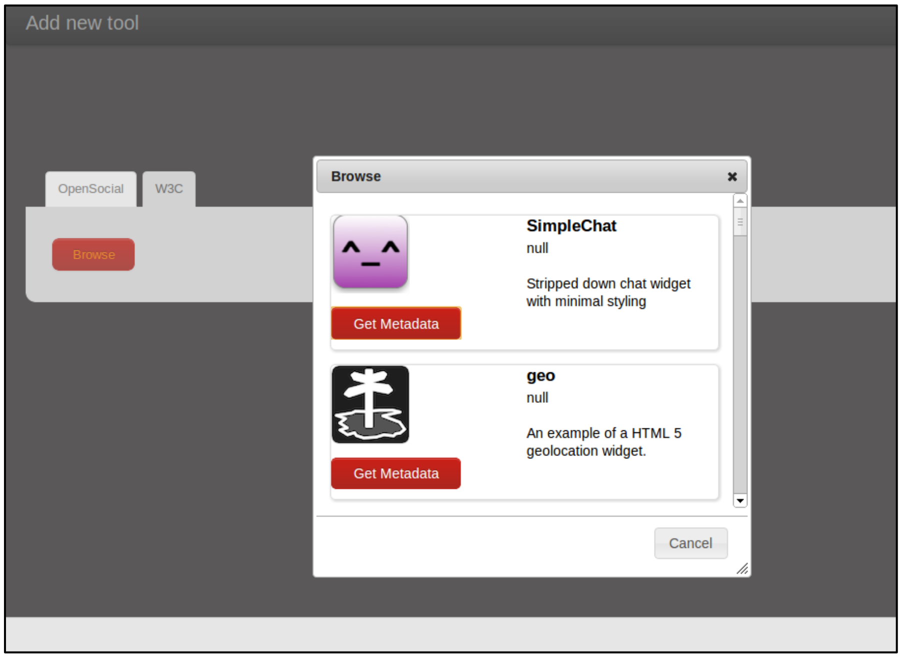

Add new tool
OpenSocial
W3C
Browse
Browse
SimpleChat
null
Stripped down chat widget with minimal styling
Get Metadata
geo
null
An example of a HTML 5 geolocation widget.
Get Metadata
Cancel

Figure 11: Adding a new W3C widget

2.2.2. Adding a New Tab

1. A new Tab can be added by clicking the '+' mark at the top of the existing Tab.
2. Tab can be named and given a preferred layout from the given options using the pop window.

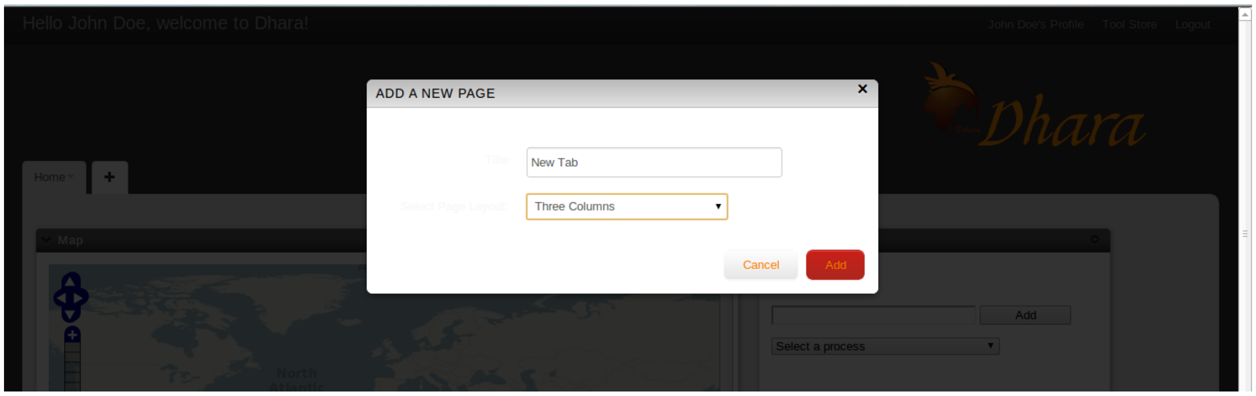


Figure 12: Adding a new Tab

2.2.3. Using Dhara – tools

1. Click on the arrow mark next to the name of the tool to collapse and reopen the tool
2. Click on the settings sign on the right corner of the tool title bar to open the panel of settings for each tool (delete, maximize, about etc.)

2.2.3.1. Map tool

1. The zoom settings are provided on the left side of the map
2. The set of 4 arrows on top of zoom panel can be used to pan the map.
3. Tool set on top right corner of the map (from left to right)
    - select a feature(polygons, lines, points) drawn on the map
    - draw polygons on the map
    - draw lines on the map
    - draw points on the map
    - pan the map
4. Click on the top right side '+' button to open the panel of layers. Clicking on the '-' sign once the panel is open will close it.
    - Base layer: The layer to be visible on the map can be selected using radio buttons.
    - Overlays: The layers to be visible on the map can be selected using the check boxes. By default only the scratchpad layer is added. (The shapes you draw on the map are added to the scratch pad layer. The data loaded through other tools (WFS, GML, KML) will be added to a new layer each time.)
5. Click on the bottom right side '+' button to open the layer to select the focused area on the map. Clicking on the '-' sign once the panel is open will close it.

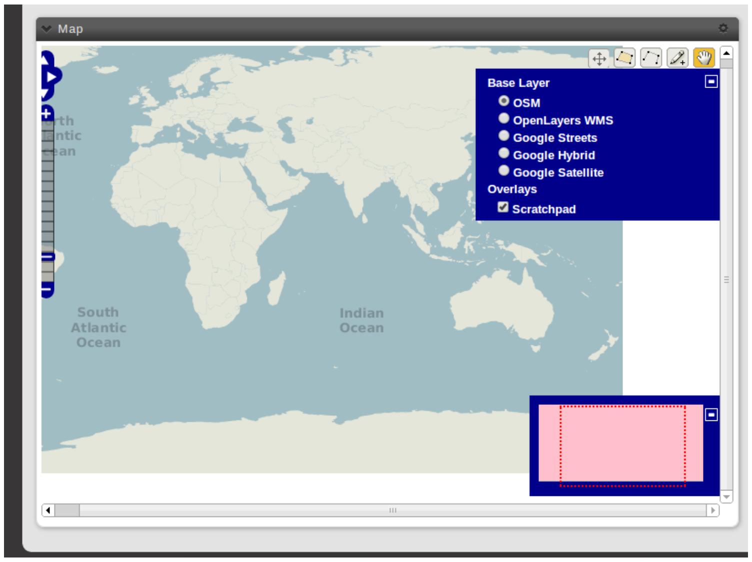


Figure 13: Map Tool

2.2.3.2. WPS Client Tool

- To access a WPS server through the WPS Client

1. Add a URL of a WPS server instance in the text area provided.
   Sample WPS server instance:
   http://54.200.198.155:8090/52n-wps-webapp-3.3.0-SNAPSHOT/WebProcessingService
2. Click 'Add' button.
3. Select a WPS process to be used from the drop down list.

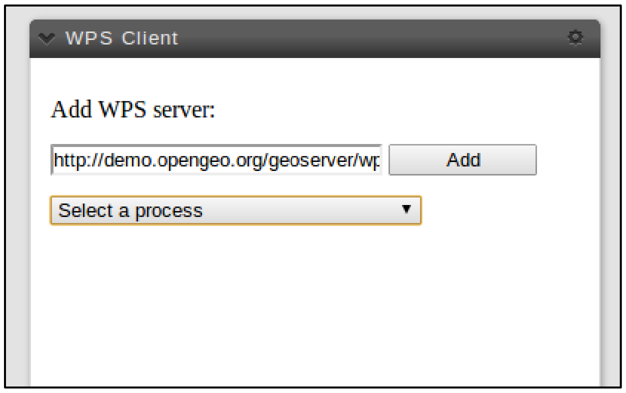
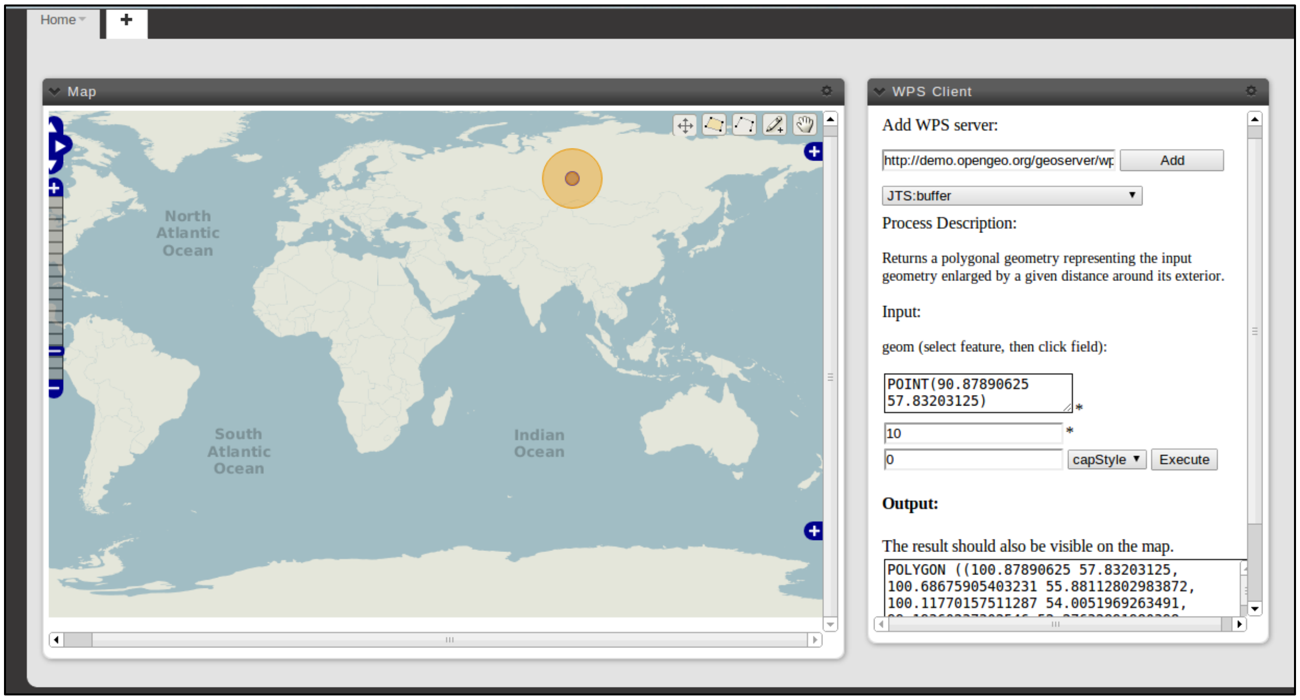


Figure 14: Adding a WPS server instance

- To use the selected WPS process

1. Once the process is selected from the list, Process description and Input fields will appear.
   Sample WPS process*: JTS:buffer*

2. The input for the 'geom' field takes the geometry on which the process should work on. The input can be either provided as text in WKT form by typing or by selecting a feature from map.
   POINT(90.87890625 57.83203125)
3. Draw the geometry on map using the tool set (polygon, line, point)
4. Select the drawn feature using the select tool.
5. Then click on the 'geom' input area. The selected geometry will be added in WKT format.
6. Fill the rest of the input fields according to the description.
7. Click execute button
8. The output will be shown in the client tool and/or on the map depending on the process.



Add WPS server:

http://demo.opengeo.org/geoserver/wp

JTS:buffer

Process Description:

Returns a polygonal geometry representing the input geometry enlarged by a given distance around its exterior.

Input:

geom (select feature, then click field):

```
POINT(90.87890625
57.83203125)
```

10 *

0 capStyle

Output:

The result should also be visible on the map.

```
POLYGON ((100.87890625 57.83203125,
100.68675905403231 55.88112802983872,
100.11770157511287 54.0051969263491,
```

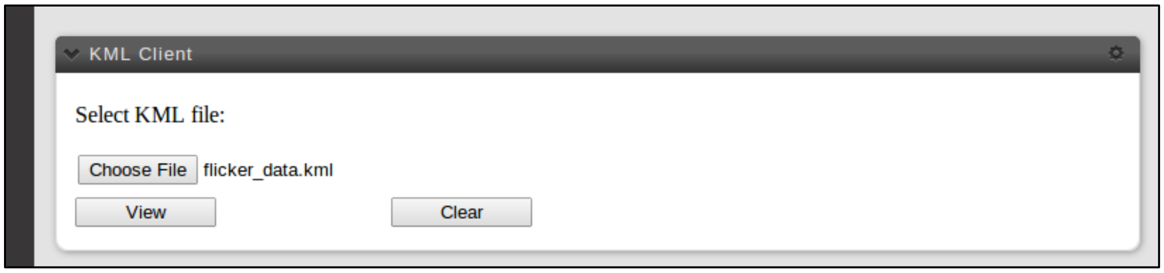
Figure 15: WPS process executed

2.2.3.3. WFS Client Tool

1. Add a URL of a WFS server instance to the text area provided.
   Sample WFS server instance:
   http://54.200.198.155:8094/geoserver
2. Click 'Add' button
3. The URL will appear as the current service. You can select other server instances you previously accessed from the drop down list.
4. Select the feature to visualize from the 'feature' drop down list.
5. Click ' Add to map' to visualize the feature on map.

2.2.3.4. KML Client

1. Click 'Browse' button to select a KML (.kml) file from the file system.
2. Click 'Add to map' button to visualize the selected file on the map.



Select KML file:

flicker_data.kml

Figure 16: KML Client

2.2.3.5. GML Client

1. Click 'Browse' button to select a GML (.xml) file from the file system.
2. Click 'Add to map' button to visualize the selected file on the map.

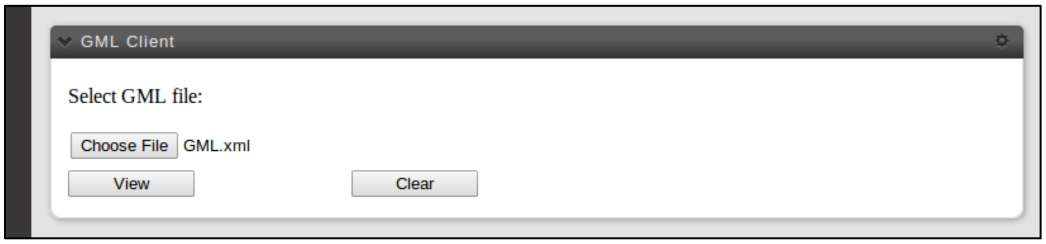


Figure 17: GML Client

3. **Test Suite**
   - Dhara has a test suite for test the complete portal functionality.
   - User first needs to cloned the test suit in Github to launch the test suite
   - Then test suite has a configuration file where user needs to specify the complete portal setup.
   - Then executing “**mvn clean install**” inside the test suite will run all the test cases of the portal
   - Test suite includes tests for test, Apache Airavata API; 52o North WPS class deployment, Portal functions and backend web application REST API.

Example:

Configurations file for testing in local deployment

```xml
<?xml version="1.0" encoding="UTF-8"?>
<server-configurations>
    <airavata-configuration>
        <server>
            <username>admin</username>
            <password>admin</password>
            <message-box></message-box>
            <gateway-name>default</gateway-name>
            <port>8081</port>
            <server-context>airavata-registry</server-context>
            <server-url>localhost</server-url>
            <broker></broker>
            <jcr></jcr>
            <gfac></gfac>
        </server>
    </airavata-configuration>
    <backend-configuration>
        <server>
            <server-url>http://localhost:8090/portal</server-url>
            <password>admin</password>
            <username>admin</username>
        </server>
    </backend-configuration>
    <wps-52north-configuration>
        <server>
            <username>admin</username>
            <password>admin</password>
            <server-url>http://localhost:8090/52n-wps-webapp-3.3.0-SNAPSHOT/webAdmin/DynamicDeployProcesstest.jsp</server-url>
            <test-url>http://localhost:8090/52n-wps-webapp-3.3.0-SNAPSHOT/webAdmin/DynamicDeployProcesstest.jsp</test-url>
        </server>
    </wps-52north-configuration>
    <portal-configuration>
        <server>
            <username>canonical</username>
            <password>canonical</password>
            <server-url>http://localhost:8080/portal</server-url>
        </server>
    </portal-configuration>
</server-configurations>
```