

eXo Platform 3.0 - Social Functions

Copyright ©

1. Applications	1
1.1. List of Portlets in Social	1
1.2. List of Gadgets in Social	1
2. Configuration	3
2.1. RelationshipManager	3
2.2. SpaceIdentityProvider	3
2.3. SpaceApplicationHandler	3
2.4. ExoPeopleService	3
2.5. Space Service	3
2.5.1. Description	3
2.5.2. Components configuration	3
2.5.3. External plug-in configuration	4
2.6. Activity Manager	4
2.6.1. Description	4
2.6.2. Component plug-in configuration	5
2.6.3. External plug-in configuration	5
2.7. Identity Manager	6
2.7.1. Description	6
2.7.2. Component plug-in configuration	6
2.8. Spaces Template configuration	6
2.9. Configure the oauth 2 legged scenario	8
2.9.1. Generate the certificates	8
2.9.2. Configure the property file	8
3. Developers References	9
3.1. UI Extensions	9
3.1.1. About Activity Plug-in	9
3.1.2. Create the activity plug-in	9
3.1.3. Summary	17
3.2. Overridable Components	17
3.3. Plug-in SPI	18
3.4. Public REST APIs	18
3.4.1. Service name and description	18
3.4.2. Location	21
3.5. Public Javascript APIs	21

Chapter 1. Applications

1.1. List of Portlets in Social

All the portlets are in the file *social-portlet.war*

Portlet name	Description
Members Portlet	To enable users to search for Space members or list Space members in the alphabet order.
My Spaces Portlet	To display the Space page where you can add new Spaces, search for Spaces or list Spaces in the alphabet order.
Space Activity Stream Portlet	To share Spaces activity information.
Invitations Portlet	To list all people that invite users.
Requests Portlet	To list all invitations that the user requests.
Invitation Spaces Portlet	To enable users to search for Spaces, and list spaces in the alphabet order and view all the Space invitations.
Pending Spaces Portlet	To display the Space requests to join page where you can search for Spaces, list spaces in the alphabet order and view all the pending requests to join Spaces.
Public Spaces Portlet	To display the Public Spaces page where you can search for Spaces, list Spaces in the alphabet order and view all available public Spaces to join.
User Activity Stream Portlet	To update and share user's activities and/or status.
People Portlet	To display the People page where you can find people, display people names in the alphabet order.
Connections Portlet	To display the Connections page where you can access your network, invitations to connect or connection requests.
User Profile Portlet	To display the user profile page where users can view and/or edit basic information, contacts, experience or change their avatar.

1.2. List of Gadgets in Social

All Social Gadgets are in the *opensocial.war* file

Gadget name	Used REST service	Description	
Activities	ActivitiesRestServices	To manage activities of users. For example, update status, like/unlike activities, comment activities, delete activities and delete comments.	
ApplicationList	AppsRestService	To get information of all existing applications (portlet type) in the Portal. The gadget is used for the demo rest service to get application from the portal container.	
RSSFetch	N/A	The number of rss per page is 10 by default; Get the rss from input url (Default url is http://blog.exoplatform.org/feed/)	
UserSpaces	SpacesRestService		
MySpaces	SpacesRestService	To get information of spaces, including my space and pending spaces of current login users. This gadget is used for the demo using tab in writing gadgets.	

Chapter 2. Configuration

2.1. RelationshipManager

The Service is used to manipulate user relationships.

2.2. SpaceIdentityProvider

The provider is to give identity space instances.

2.3. SpaceApplicationHandler

The service is to handle all events related to creating and managing all the application spaces.

2.4. ExoPeopleService

The service is to manipulate all data related to people in the Portal.

2.5. Space Service

2.5.1. Description

The service for spaces management includes creating spaces, and installing applications.

2.5.2. Components configuration

Component name	Description
SpaceServiceImpl	Implementation class of Space Service.

```
<component>
  <key>org.exoplatform.social.core.space.spi.SpaceService</key>
  <type>org.exoplatform.social.core.space.impl.SpaceServiceImpl</type>
  <init-params>
    <!-- Configure the applications to install in a space -->
    <values-param>
      <name>space.homeNodeApp</name>
      <value>SpaceActivityStreamPortlet</value>
    </values-param>
    <!-- Configure removable application or not <value>Application:removable</value> -->
    <values-param>
      <name>space.apps</name>
      <value>DashboardPortlet:true</value>
      <value>SpaceSettingPortlet:false</value>
      <value>MembersPortlet:true</value>
    </values-param>
  </init-params>
</component>
```

Configuration name	Data Type	Possible value	Default Value	Description
SpaceActivityStreamPortlet	Portlet	N/A	SpaceActivityStreamPortlet	The name of portlet displaying activities of spaces
space.apps	String list	Portlets' name: true/false	DashboardPortlet:true SpaceSettingPortlet:true MembersPortlet:true	The list of configurations for portlets used as portlet applications.

2.5.3. External plug-in configuration

```
<external-component-plugins>
  <target-component>org.exoplatform.social.core.space.spi.SpaceService</target-component>
  <component-plugin>
    <name>portlets.prefs.required</name>
    <set-method>setPortletsPrefsRequired</set-method>
    <type>org.exoplatform.social.core.application.PortletPreferenceRequiredPlugin</type>
    <init-params>
      <values-param>
        <name>portletsPrefsRequired</name>
        <value>SpaceActivityStreamPortlet</value>
        <value>SpaceSettingPortlet</value>
        <value>MembersPortlet</value>
      </values-param>
    </init-params>
  </component-plugin>
</external-component-plugins>
```

In which:

Name	Set-method	Type	Description
PortletPreferenceRequiredPlugin	setPortletsPrefsRequired	org.exoplatform.social.core.application.PortletPreferenceRequiredPlugin	Configuration of portlets which will have portlet preference of space context.

- Init-params:

Name	Possible value	Default Value	Description
portletsPrefsRequired	Portlet names	SpaceActivityStreamPortlet; SpaceSettingPortlet; MembersPortlet	List of portlets which need to be saved and get the space context name.

2.6. Activity Manager

2.6.1. Description

The service is to manipulate space and user activities.

2.6.2. Component plug-in configuration

Exception occurred, see logs

In which,

Name	Set-method	Type	Description
OSHtmlSanitizerProcessor	addProcessorPlugin	org.exoplatform.social.core.processor.OSHtmlSanitizerProcessor	The <code>OSHtmlSanitizerProcessor</code> valid html tags appearing in the Activity body (content).

- Init-params:

Name	Possible value	Default Value	Description
allowedTags	html tags	b, i, a, span, em, strong, p, ol, ul, li, br, img	To process and render html tags in the activity content (body).

2.6.3. External plug-in configuration

```
<external-component-plugins>
  <target-component>org.exoplatform.social.core.manager.ActivityManager</target-component>
  <component-plugin>
    <name>MentionsProcessor</name>
    <set-method>addProcessorPlugin</set-method>
    <type>org.exoplatform.social.core.processor.MentionsProcessor</type>
    <init-params>
      <value-param>
        <name>priority</name>
        <description>priority of this processor (lower number will be executed first)</description>
        <value>2</value>
      </value-param>
    </init-params>
  </component-plugin>
</external-component-plugins>
```

In which:

Name	Set-method	Type	Description
MentionsProcessor	addProcessorPlugin	org.exoplatform.social.core.processor.MentionsProcessor	The <code>MentionsProcessor</code> substitutes <code>@username</code> expressions by a link on the user profile.

- Init-params:

Name	Possible value	Default Value	Description
priority	priority number	2	Priority of this processor (The lower level will be executed first).

2.7. Identity Manager

2.7.1. Description

The service is to manipulate the identity operations like creating, getting, deleting or finding a profile.

2.7.2. Component plug-in configuration

```
<component>
  <key>org.exoplatform.social.core.manager.IdentityManager</key>
  <type>org.exoplatform.social.core.manager.IdentityManager</type>
  <component-plugins>
    <component-plugin>
      <name>SpaceIdentityProvider plugin</name>
      <set-method>registerIdentityProviders</set-method>
      <type>org.exoplatform.social.core.identity.IdentityProviderPlugin</type>
      <init-params>
        <values-param>
          <name>providers</name>
          <description>Identity Providers</description>
          <value>org.exoplatform.social.core.identity.provider.SpaceIdentityProvider</value>
        </values-param>
      </init-params>
    </component-plugin>
  </component-plugins>
</component>
```

In which:

Name	Set-method	Type	Description
SpaceIdentityProvider	registerIdentityProviders	org.exoplatform.social.core.identity.IdentityProviderPlugin	The plugin that provides identity for a space

- Init-params:

Name	Possible value	Default Value	Description
providers	Every other identity providers	org.exoplatform.social.core.identity.provider.SpaceIdentityProvider	Identity provider instances for managing identities.

2.8. Spaces Template configuration

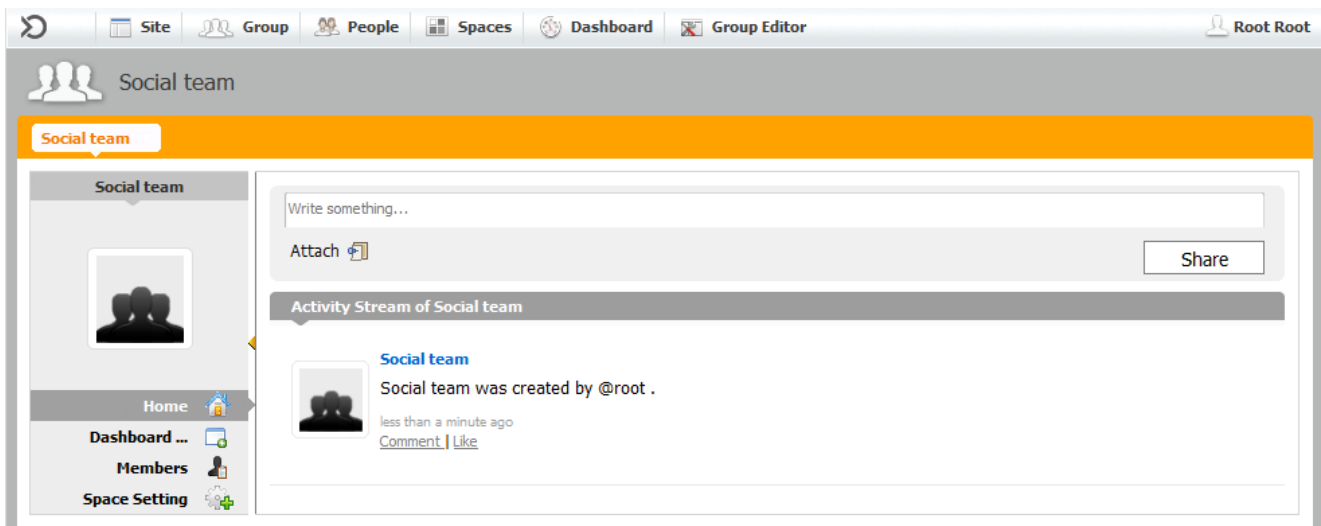
In the eXo Spaces, we may have two space types (classic and webos spaces). This is for the classic mode (it's the only one implemented now).

For the classic space, we can pre-configure the template, meaning that you can set up where your **menu** will be displayed or where your **application** will be displayed.

Here is an example of configuration file that displays the menu on the left. The Application will be inserted in the container with the id **Application**:

```
<page>
  <owner-type/>
  <owner-id/>
  <name/>
  <container id="SpacePage" template="system:/groovy/portal/webui/container/UITableColumnContainer.gtmpl">
    <container id="Menu" template="system:/groovy/portal/webui/container/UIContainer.gtmpl">
      <portlet-application>
        <portlet>
          <application-ref>space</application-ref>
          <portlet-ref>SpaceMenuPortlet</portlet-ref>
        </portlet>
        <access-permissions>*:/platform/users</access-permissions>
        <show-info-bar>false</show-info-bar>
      </portlet-application>
    </container>
    <container id="Application" template="system:/groovy/portal/webui/container/UIContainer.gtmpl">
      </container>
    </container>
  </page>
```

In this example, the outer container contains two inner containers: one container has id as **Menu** for your Menu and another has id as **Application** containing your applications.



If you want to put your menu in right and your application in left, you can swap the declared position of two containers:

```
<page>
  <owner-type/>
  <owner-id/>
  <name/>
  <container id="SpacePage" template="system:/groovy/portal/webui/container/UITableColumnContainer.gtmpl">
    <container id="Application" template="system:/groovy/portal/webui/container/UIContainer.gtmpl">
      <portlet-application>
        <portlet>
          <application-ref>space</application-ref>
          <portlet-ref>SpaceMenuPortlet</portlet-ref>
        </portlet>
      </portlet-application>
    </container>
    <container id="Menu" template="system:/groovy/portal/webui/container/UIContainer.gtmpl">
      </container>
    </container>
  </page>
```

```
</portlet>
<access-permissions>*:/platform/users</access-permissions>
<show-info-bar>false</show-info-bar>
</portlet-application>
</container>
</container>
</container>
</page>
```

Here is the configure file in FishEye:
<http://fisheye.exopatform.org/browse/social/trunk/extension/war/src/main/webapp/WEB-INF/conf/portal/template/pages/sp>

In your tomcat, this configuration file is at
\$EXOTOMCAT/webapps/social-ext/WEB-INF/conf/portal/template/pages/space/page.xml.

2.9. Configure the oauth 2 legged scenario

This is to explain how to configure the oAuth 2 legs scenario in openSocial.

For more information about this, visit the website: [great article](#)

2.9.1. Generate the certificates

To generate the key:

```
$ openssl req -newkey rsa:1024 -days 365 -nodes -x509 -keyout testkey.pem
  -out testkey.pem -subj '/CN=mytestkey'
$ openssl pkcs8 -in testkey.pem -out oauthkey.pem -topk8 -nocrypt -outform PEM
```

2.9.2. Configure the property file

Edit container.js and change the following parameter to point to your private key and your key name.

```
"gadgets.signingKeyFile" : "oauth.pem",
"gadgets.signingKeyName" : "oauthKey",
```

Chapter 3. Developers References

3.1. UI Extensions

3.1.1. About Activity Plug-in

Since the Social 1.1.0, the activity plug-in feature was introduced that enables the activity composer extension and custom UI component to be used for displaying one activity by its type.

3.1.2. Create the activity plug-in

At first, you should have an idea about the UI Extension Framework. If you have already worked with the UI Extension Framework, it's really easy to create the activity plug-in. If no, you have chance to work with it visiting [UI Extension Framework](#).

3.1.2.1. Create a custom UI component for displaying the activity based on its type.



Note

("Project Code can be downloadable [here](#)")

When an activity is displayed, UIActivityFactory will look for its registered custom activity display by activity's type. If not found, UIDefaultActivity will be called for displaying that activity.

For example, in Social, there is an activity of type "exosocial:spaces" created by SpaceActivityPublisher. If you want to display it with our own UI component instead of default ones.

First, create a sample project:

```
mvn archetype:generate
[INFO] Scanning for projects...
[INFO] Searching repository for plugin with prefix: 'archetype'.
[INFO] artifact org.apache.maven.plugins:maven-archetype-plugin: checking for updates from central
[INFO] snapshot org.apache.maven.plugins:maven-archetype-plugin:2.0-alpha-6-SNAPSHOT: checking for updates from central
[INFO] snapshot org.apache.maven.archetype:maven-archetype:2.0-alpha-6-SNAPSHOT: checking for updates from central
[INFO] -----
[INFO] Building Maven Default Project
[INFO]   task-segment: [archetype:generate] (aggregator-style)
[INFO] -----
[INFO] Preparing archetype:generate
[INFO] No goals needed for project - skipping
[INFO] snapshot org.apache.maven.archetype:archetype-common:2.0-alpha-6-SNAPSHOT: checking for updates from central
[INFO] snapshot org.apache.maven.archetype:archetype-common:2.0-alpha-6-SNAPSHOT: checking for updates from apac
[INFO] [archetype:generate {execution: default-cli}]
[INFO] Generating project in Interactive mode
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.archetypes:maven-archetype-quick
Choose archetype:
1: remote -> docbkx-quickstart-archetype (null)
2: remote -> gquery-archetype (null)
3: remote -> gquery-plugin-archetype (null)
//....

285: remote -> wicket-scala-archetype (Basic setup for a project that combines Scala and Wicket,
        depending on the Wicket-Scala project. Includes an example Specs
        test.)
286: remote -> circumflex-archetype (null)
Choose a number: 76: 76
Choose version:
```

```

1: 1.0
2: 1.0-alpha-1
3: 1.0-alpha-2
4: 1.0-alpha-3
5: 1.0-alpha-4
6: 1.1
Choose a number: : 1
Define value for property 'groupId': : org.exoplatform.social.samples
Define value for property 'artifactId': : exo.social.samples.activity-plugin
Define value for property 'version': 1.0-SNAPSHOT: 1.0.0-SNAPSHOT
Define value for property 'package': org.exoplatform.social.samples: org.exoplatform.social.samples.activityPlugin
Confirm properties configuration:
groupId: org.exoplatform.social.samples
artifactId: exo.social.samples.activity-plugin
version: 1.0.0-SNAPSHOT
package: org.exoplatform.social.samples.activityPlugin
Y: y

```

Edit the pom.xml file as follows. Note that we will work with the latest Social release: 1.1.0-CR01.

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.exoplatform.social</groupId>
    <artifactId>social-project</artifactId>
    <version>1.1.0-CR01</version>
  </parent>

  <groupId>org.exoplatform.social.samples</groupId>
  <artifactId>exo.social.samples.activity-plugin</artifactId>
  <packaging>jar</packaging>
  <version>1.1.0-CR01</version>

  <name>exo.social.samples.activity-plugin</name>

  <build>
    <sourceDirectory>src/main/java</sourceDirectory>
    <outputDirectory>target/classes</outputDirectory>
    <resources>
      <resource>
        <directory>src/main/resources</directory>
        <includes>
          <include>**/*.gtmpl</include>
        </includes>
      </resource>
      <resource>
        <directory>src/main/java</directory>
        <includes>
          <include>**/*.xml</include>
        </includes>
      </resource>
    </resources>
  </build>

  <dependencies>

    <dependency>
      <groupId>org.exoplatform.portal</groupId>
      <artifactId>exo.portal.webui.core</artifactId>
      <scope>provided</scope>
    </dependency>

    <dependency>
      <groupId>org.exoplatform.portal</groupId>
      <artifactId>exo.portal.webui.portal</artifactId>
      <scope>provided</scope>
    </dependency>

    <dependency>
      <groupId>org.exoplatform.social</groupId>
      <artifactId>exo.social.component.core</artifactId>
      <scope>provided</scope>
    </dependency>
  </dependencies>

```

```

<dependency>
  <groupId>org.exoplatform.social</groupId>
  <artifactId>exo.social.component.webui</artifactId>
  <scope>provided</scope>
</dependency>

<dependency>
  <groupId>org.exoplatform.social</groupId>
  <artifactId>exo.social.component.service</artifactId>
  <scope>provided</scope>
</dependency>

</dependencies>

</project>

```

To use the custom UI component for displaying its activity, we need a subclass of BaseUIActivity. We call this UISpaceSimpleActivity:

```

package org.exoplatform.social.samples.activityplugin;

import org.exoplatform.social.webui.activity.BaseUIActivity;
import org.exoplatform.webui.config.annotation.ComponentConfig;
import org.exoplatform.webui.core.lifecycle.UIFormLifecycle;

/**
 * Created by The eXo Platform SAS
 * Author : eXoPlatform
 *         exo@exoplatform.com
 * Aug 23, 2010
 */
@ComponentConfig(
  lifecycle = UIFormLifecycle.class,
  template = "classpath:groovy/social/plugin/space/UISpaceSimpleActivity.gtmpl"
)
public class UISpaceSimpleActivity extends BaseUIActivity {
}

```

The template UISpaceSimpleActivity.gtmpl should be created under main/resources/groovy/social/plugin/space:

```

<div>This is a space activity ui component displayed for type "exosocial:spaces"</div>

```

An activity builder is also needed which will be explained later.

```

package org.exoplatform.social.samples.activityplugin;

import org.exoplatform.social.core.activity.model.Activity;
import org.exoplatform.social.webui.activity.BaseUIActivity;
import org.exoplatform.social.webui.activity.BaseUIActivityBuilder;

/**
 * Created by The eXo Platform SAS
 * Author : eXoPlatform
 *         exo@exoplatform.com
 * Aug 19, 2010
 */
public class SimpleSpaceUIActivityBuilder extends BaseUIActivityBuilder {

  @Override
  protected void extendUIActivity(BaseUIActivity uiActivity, Activity activity) {
    // TODO Auto-generated method stub
  }

}

```

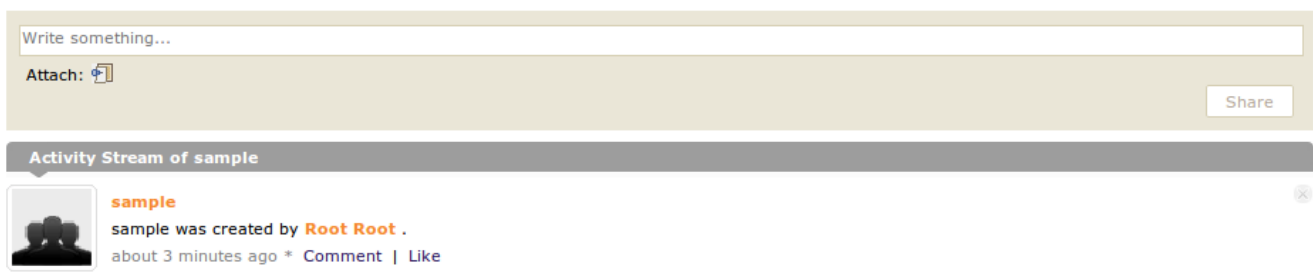
Next, create configuration.xml under conf/portal:

```
<configuration xmlns="http://www.exoplaform.org/xml/ns/kernel_1_1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema"
  <external-component-plugins>
    <target-component>org.exoplatform.webui.ext.UIExtensionManager</target-component>
    <component-plugin>
      <name>add.action</name>
      <set-method>registerUIExtensionPlugin</set-method>
      <type>org.exoplatform.webui.ext.UIExtensionPlugin</type>
      <init-params>
        <object-param>
          <name>Simple Space Activity</name>
          <object type="org.exoplatform.social.webui.activity.UIActivityExtension">
            <field name="type"><string>org.exoplatform.social.webui.activity.BaseUIActivity</string></field>
            <field name="name"><string>exosocial:spaces</string></field>
            <field name="component"><string>org.exoplatform.social.samples.activityplugin.UISpaceSimpleActivity</string></field>
            <field name="activityBuiderClass"><string>org.exoplatform.social.samples.activityplugin.SimpleSpace</string></field>
          </object>
        </object-param>
      </init-params>
    </component-plugin>
  </external-component-plugins>
</configuration>
```

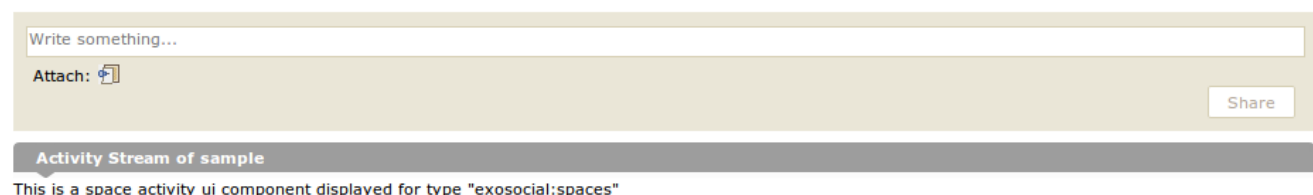
Note that exosocial:spaces, and its value must match the activity's type you want to display with your UI component.

Assume that we have already built the Social project version 1.1.0-CR01 or above. If you don't know how to, have a look at [Building from Sources](#) with [Social 1.1.0-CR01](#). Build sample project and copy jar file to /tomcat/lib. Run Social, create a space and access it. You can see:

space's activity of type "exosocial:spaces" is displayed by default in Social:



With our custom UI component for displaying activity of type: "exosocial:spaces":



1.1.1 Make the custom UI activity display have the look and feel and function like default one.

When displaying an activity, we should make sure the look and feel of the custom UI component is consistent and match other activities and have the functions of like, comments. So, to create the another UI component to display, we call `UISpaceLookAndFeelActivity`:

```
package org.exoplatform.social.samples.activityplugin;
```

```

import org.exoplatform.social.webui.activity.BaseUIActivity;
import org.exoplatform.webui.config.annotation.ComponentConfig;
import org.exoplatform.webui.config.annotation.EventConfig;
import org.exoplatform.webui.core.lifecycle.UIFormLifecycle;

/**
 * Created by The eXo Platform SAS
 * Author : eXoPlatform
 *         exo@exoplatform.com
 * Aug 23, 2010
 */
@ComponentConfig(
    lifecycle = UIFormLifecycle.class,
    template = "classpath:groovy/social/plugin/space/UISpaceLookAndFeelActivity.gtmpl",
    events = {
        @EventConfig(listeners = BaseUIActivity.ToggleDisplayLikesActionListener.class),
        @EventConfig(listeners = BaseUIActivity.ToggleDisplayCommentFormActionListener.class),
        @EventConfig(listeners = BaseUIActivity.LikeActivityActionListener.class),
        @EventConfig(listeners = BaseUIActivity.SetCommentListStatusActionListener.class),
        @EventConfig(listeners = BaseUIActivity.PostCommentActionListener.class),
        @EventConfig(listeners = BaseUIActivity.DeleteActivityActionListener.class, confirm = "UIActivity.msg.Are_You_Sure_To_Delete_Activity"),
        @EventConfig(listeners = BaseUIActivity.DeleteCommentActionListener.class, confirm = "UIActivity.msg.Are_You_Sure_To_Delete_Comment")
    }
)
public class UISpaceLookAndFeelActivity extends BaseUIActivity {
}

```

Now, create the `UISpaceLookAndFeelActivity` template by copying the content of `UIDefaultActivity.gtmpl` to this template file:

```

<%
import org.exoplatform.portal.webui.util.Util;
import org.exoplatform.webui.form.UIFormTextAreaInput;

def pcontext = Util.getPortalRequestContext();
def jsManager = pcontext.getJavascriptManager();
def labelActivityHasBeenDeleted = _ctx.appRes("UIActivity.label.Activity_Has_Been_Deleted");
def activity = uicomponent.getActivity();
def activityDeletable = uicomponent.isActivityDeletable();
%>

<% if (activity) { //process if not null

    jsManager.importJavascript("eXo.social.Util", "/social-resources/javascript");
    jsManager.importJavascript("eXo.social.PortalHttpRequest", "/social-resources/javascript");
    jsManager.importJavascript("eXo.social.webui.UIForm", "/social-resources/javascript");
    jsManager.i!mportJavascript("eXo.social.webui.UIActivity", "/social-resources/javascript");

    def labelComment = _ctx.appRes("UIActivity.label.Comment");
    def labelLike = _ctx.appRes("UIActivity.label.Like");
    def labelUnlike = _ctx.appRes("UIActivity.label.Unlike");
    def labelSource = _ctx.appRes("UIActivity.label.Source");
    def inputWriteAComment = _ctx.appRes("UIActivity.input.Write_A_Comment");
    def labelShowAllComments = _ctx.appRes("UIActivity.label.Show_All_Comments");
    def labelHideAllComments = _ctx.appRes("UIActivity.label.Hide_All_Comments");
    def labelOnePersonLikeThis = _ctx.appRes("UIActivity.label.One_Person_Like_This");
    def labelPeopleLikeThis = _ctx.appRes("UIActivity.label.People_Like_This");
    def labelYouLikeThis = _ctx.appRes("UIActivity.label.You_Like_This");
    def labelYouAndOnePersonLikeThis = _ctx.appRes("UIActivity.label.You_And_One_Person_Like_This");
    def labelYouAndPeopleLikeThis = _ctx.appRes("UIActivity.label.You_And_People_Like_This");

    def likeActivityAction = uicomponent.event("LikeActivity", "true");
    def unlikeActivityAction = uicomponent.event("LikeActivity", "false");

    def commentList = uicomponent.getComments();
    def allComments = uicomponent.getAllComments();
    if (allComments) {
        labelShowAllComments = labelShowAllComments.replace("{0}", allComments.size() + "");
        labelHideAllComments = labelHideAllComments.replace("{0}", allComments.size() + "");
    }
    def displayedIdentityLikes = uicomponent.getDisplayedIdentityLikes();
    def identityLikesNum = 0;
}

```

```

def labelLikes = null;
def toggleDisplayLikesAction = uicomponent.event("ToggleDisplayLikes");
def startTag = "<a onclick=\"$toggleDisplayLikesAction\" id=\"ToggleDisplayListPeopleLikes${activity.id}\" href=\"#\"";
def endTag = "</a>";
if (displayedIdentityLikes != null) {
    identityLikesNum = displayedIdentityLikes.length;
}
def commentListStatus = uicomponent.getCommentListStatus();
def commentFormDisplayed = uicomponent.isCommentFormDisplayed();
def likesDisplayed = uicomponent.isLikesDisplayed();
//params for init UIActivity javascript object
def params = ""
    {activityId: '${activity.id}',
      inputWriteAComment: '$inputWriteAComment',
      commentMinCharactersAllowed: ${uicomponent.getCommentMinCharactersAllowed()},
      commentMaxCharactersAllowed: ${uicomponent.getCommentMaxCharactersAllowed()},
      commentFormDisplayed: $commentFormDisplayed,
      commentFormFocused: ${uicomponent.isCommentFormFocused()}
    }
""
jsManager.addOnLoadJavascript("initUIActivity${activity.id}");
//make sures commentFormFocused is set to false to prevent any refresh to focus, only focus after post a comment
uicomponent.setCommentFormFocused(false);
def activityUserName, activityUserProfileUri, activityImageSource, activityContentBody, activityPostedTime;
def commentFormBlockClass = "", listPeopleLikeBlockClass = "", listPeopleBGClass = "";
if (!commentFormDisplayed) {
    commentFormBlockClass = "DisplayNone";
}

if (!likesDisplayed) {
    listPeopleLikeBlockClass = "DisplayNone";
}

if (uicomponent.isLiked()) {
    if (identityLikesNum > 1) {
        labelLikes = labelYouAndPeopleLikeThis.replace("{start}", startTag).replace("{end}", endTag).replace("{0}", identityLikesNum);
    } else if (identityLikesNum == 1) {
        labelLikes = labelYouAndOnePersonLikeThis.replace("{start}", startTag).replace("{end}", endTag);
    } else {
        labelLikes = labelYouLikeThis;
    }
} else {
    if (identityLikesNum > 1) {
        labelLikes = labelPeopleLikeThis.replace("{start}", startTag).replace("{end}", endTag).replace("{0}", identityLikesNum);
    } else if (identityLikesNum == 1) {
        labelLikes = labelOnePersonLikeThis.replace("{start}", startTag).replace("{end}", endTag);
    }
}

if (!labelLikes) {
    //hides diplayPeopleBG
    listPeopleBGClass = "DisplayNone";
}

activityContentTitle = activity.title;
activityPostedTime = uicomponent.getPostedTimeString(activity.postedTime);
activityUserName = uicomponent.getUserFullName(activity.userId);
activityUserProfileUri = uicomponent.getUserProfileUri(activity.userId);

activityImageSource = uicomponent.getUserAvatarImageSource(activity.userId);
if (!activityImageSource) {
    activityImageSource = "/social-resources/skin/ShareImages/SpaceImages/SpaceLogoDefault_61x61.gif";
}

%>

<div class="UIActivity">
    <script type="text/javascript">
        function initUIActivity${activity.id}() {
            new exo.social.webui.UIActivity($params);
        }
    </script>

    <% uiForm.begin() %>
    <div class="NormalBox clearfix">
        <a class="Avatar" title="$activityUserName" href="$activityUserProfileUri">
            
        </a>

```

```

<div class="ContentBox" id="ContextBox${activity.id}">
  <div class="TitleContent clearfix">
    <div class="Text">
      <a title="$activityUserName" href="$activityUserProfileUri">$activityUserName</a>
    </div>
    <% if (activityDeletable) {%>
      <div onclick="<%= uicomponent.event("DeleteActivity", uicomponent.getId(), ""); %>" class="CloseContentE
    <%}%>
  </div>
  <div class="Content">
    $activityContentTitle (from custom ui component)<br>
  </div>
  <div class="LinkCM">
    <span class="DateTime">$activityPostedTime *</span>
    <% def toggleDisplayCommentAction = uicomponent.event('ToggleDisplayCommentForm', null, false);
      def commentLink = "";
    %>
    <a class="LinkCM $commentLink" onclick="$toggleDisplayCommentAction" id="CommentLink${activity.id}" href=
      $labelComment
    </a> |
    <% if (uicomponent.isLiked()) { %>
      <a onclick="$unlikeActivityAction" class="LinkCM" id="UnLikeLink${activity.id}" href="#unlike">
        $labelUnlike
      </a>
    <% } else { %>
      <a onclick="$likeActivityAction" class="LinkCM" id="LikeLink${activity.id}" href="#like">
        $labelLike
      </a>
    <% } %>
  </div>
</div>

<div class="ListPeopleLikeBG $listPeopleBGClass">
  <div class="ListPeopleLike">
    <div class="ListPeopleContent">
      <% if (!labelLikes) labelLikes = ""; %>
      <div class="Title">$labelLikes</div>
      <div class="$listPeopleLikeBlockClass">
        <%
          //def personLikeFullName, personLikeProfileUri, personLikeAvatarImageSource;

          displayedIdentityLikes.each({
            personLikeFullName = uicomponent.getUserFullName(it);
            personLikeProfileUri = uicomponent.getUserProfileUri(it);
            personLikeAvatarImageSource = uicomponent.getUserAvatarImageSource(it);
            if (!personLikeAvatarImageSource) {
              personLikeAvatarImageSource = "/social-resources/skin/ShareImages/activity/AvatarPeople.gif";
            }
          %>
          <a class="AvatarPeopleBG" title="$personLikeFullName" href="$personLikeProfileUri">
            
        <% } %>
      </div>
      <div class="ClearLeft">
        <span></span>
      </div>
    </div>
  </div>
</div>

<div class="CommentListInfo">
  <% if (uicomponent.commentListToggleable()) {
    def showAllCommentsAction = uicomponent.event("SetCommentListStatus", "all");
    def hideAllCommentsAction = uicomponent.event("SetCommentListStatus", "none");
  %>
  <div class="CommentBlock">
    <div class="CommentContent">
      <div class="CommentBorder">
        <% if (commentListStatus.getStatus().equals("latest") || commentListStatus.getStatus().equals("none"))>
          <a onclick="$showAllCommentsAction" href="#show-all-comments">
            $labelShowAllComments
          </a>
        <% } else if (commentListStatus.getStatus().equals("all")) { %>
          <a onclick="$hideAllCommentsAction" href="#hide-all-comments">
            $labelHideAllComments
          </a>
        <% } %>
      </div>
    </div>
  </div>
</div>

```

```

        </div>
    </div>
</div>
<% } %>
</div>
<% if (allComments.size() > 0) { %>
    <div class="DownIconCM"><span></span></div>
<% }%>

<%
def commenterFullName, commenterProfileUri, commenterImageSource, commentMessage, commentPostedTime;
def first = true, commentContentClass;
commentList.each({
    if (first & !uicomponent.commentListToggleable()) {
        commentContentClass = "CommentContent";
        first = false;
    } else {
        commentContentClass = "";
    }
    commenterFullName = uicomponent.getUserFullName(it.userId);
    commenterProfileUri = uicomponent.getUserProfileUri(it.userId);
    commenterImageSource = uicomponent.getUserAvatarImageSource(it.userId);
    if (!commenterImageSource) {
        commenterImageSource = "/social-resources/skin/ShareImages/activity/AvatarPeople.gif";
    }
    commentMessage = it.title;
    commentPostedTime = uicomponent.getPostedTimeString(it.postedTime);
%>
<div id="CommentBlock${activity.id}" class="CommentBox clearfix">
    <a class="AvatarCM" title="$commenterFullName" href="$commenterProfileUri">
        
    </a>
    <div class="ContentBox">
        <div class="Content">
            <a href="$commenterProfileUri"><span class="Commenter">$commenterFullName</span></a><br />
            $commentMessage
        </div>
        <div class="LinkCM">
            <span class="DateTime">$commentPostedTime</span>
        </div>
    </div>
    <%
        if (uicomponent.isCommentDeletable(it.userId)) {
    %>
        <div onclick="<%= uicomponent.event("DeleteComment", uicomponent.id, it.id); %>" class="CloseCMContentHili
    <% } %>
    </div>
<% } %>

    <div class="CommentBox $commentFormBlockClass clearfix" id="CommentFormBlock${activity.id}">
        <% uicomponent.renderChild(UIFormTextAreaInput.class); %>
        <input type="button" onclick="<%= uicomponent.event("PostComment") %>" value="$labelComment" class="Commer
    </div>

</div>
<% uiForm.end() %>
</div>
<% } else { %> <!-- activity deleted -->
<div class="UIActivity Deleted">$labelActivityHasBeenDeleted</div>
<% }%>

```

You should make needed modifications for this template. We make a small modification here:

```

<div class="Content">
    $activityContentTitle (from custom ui component)<br>
</div>

```

We need to reconfigure configuration.xml:

```

<configuration xmlns="http://www.exoplaform.org/xml/ns/kernel_1_1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema
    <external-component-plugins>

```

```

<target-component>org.exoplatform.webui.ext.UIExtensionManager</target-component>
<component-plugin>
  <name>add.action</name>
  <set-method>registerUIExtensionPlugin</set-method>
  <type>org.exoplatform.webui.ext.UIExtensionPlugin</type>
  <init-params>
    <object-param>
      <name>Look And Feel Space Activity</name>
      <object type="org.exoplatform.social.webui.activity.UIActivityExtension">
        <field name="type"><string>org.exoplatform.social.webui.activity.BaseUIActivity</string></field>
        <field name="name"><string>exosocial:spaces</string></field>
        <field name="component"><string>org.exoplatform.social.samples.activityplugin.UISpaceLookAndFeelActi
        <field name="activityBuiderClass"><string>org.exoplatform.social.samples.activityplugin.SimpleSpace
      </object>
    </object-param>
  </init-params>
</component-plugin>
</external-component-plugins>
</configuration>

```

Rebuild the sample project, copy jar to tomcat/lib. Rerun the server and see the result:

Activity Stream of sample



sample
sample was created by **Root Root** . (from custom ui component)
about 34 minutes ago * [Comment](#) | [Unlike](#)

You like this.

My comment here!!!!

Comment



Note

Currently, we have to copy and paste in the template file. We will have the full control of the UI, but it is not good when there is any change in UIDefaultActivity.

3.1.2.2. Create a composer extension for composing activity on the UI composer and display it on activity stream.

3.1.3. Summary

You can see more codes for the Social built-in plug-in ([Link Composer Plugin](#) and [Document Composer Plugin](#)) in [our project code](#)

3.2. Overridable Components

There are 2 components in Social that can be overridden: Space Application Handler & Space Service

- Space Application Handler**

```

<component>
  <key>org.exoplatform.social.core.space.spi.SpaceApplicationHandler</key>
  <type>org.exoplatform.social.core.space.impl.DefaultSpaceApplicationHandler</type>
</component>

```

- Space Service

```
<component>
  <key>org.exoplatform.social.core.space.spi.SpaceService</key>
  <type>org.exoplatform.social.core.space.impl.SpaceServiceImpl</type>
  <init-params>
    <!-- Configure the applications to install in a space -->
    <values-param>
      <name>space.homeNodeApp</name>
      <value>SpaceActivityStreamPortlet</value>
    </values-param>
    <!-- Configure removable application or not <value>Application:removable</value> -->
    <values-param>
      <name>space.apps</name>
      <value>DashboardPortlet:true</value>
      <value>SpaceSettingPortlet:false</value>
      <value>MembersPortlet:true</value>
    </values-param>
  </init-params>
</component>
```

3.3. Plug-in SPI

3.4. Public REST APIs

3.4.1. Service name and description

3.4.1.1. Activities REST service

Name	Service URL	Description
ActivitiesRestService	<code>\$portalName/social/activities</code>	To provide rest services for activity applications: like/unlike; comment; delete activity.

- API:

Name	Service URL Endpoint	Description
destroyActivity	<code>restContextName/social/activities/\$activityId/likes/delete/\$userId.\$format</code>	To delete likes of activity by activityId and return the json/xml format.
showLikes	<code>restContextName/social/activities/\$activityId/likes/show/\$format</code>	To show likes of activity by activityId and return the json/xml format.
updateLike	<code>restContextName/social/activities/\$activityId/likes/update/\$userId.\$format</code>	To update like of activity by the activityId and return the json/xml format.
destroyLike	<code>restContextName/social/activities/\$activityId/likes/delete/\$userId.\$format</code>	To delete like of activity by the activityId and return the json/xml format.
showComments	<code>restContextName/social/activities/\$activityId/comments/show/\$format</code>	To show comments of activity by the activityId and return the json/xml format.

Name	Service URL Endpoint	Description
updateComment	restContextName/social/activities/\$activityId/likes/update/\$commentId.\$format	To update the comment by the json/xml format.
destroyComment	restContextName/social/activities/\$activityId/comments/destroy/\$commentId.\$format	To destroy the comment by the json/xml format.

Parameter	Expected values
format	xml/json

Example:

- <http://localhost:9090/rest/social/activities/s08d397dg6/likesdestroy/abc.xml>
- <http://localhost:9090/rest/social/activities/s08d397dg6/likesdestroy/abc.json>

3.4.1.2. Apps REST service

Name	Service URL	Description
AppsRestService	\$restContextName/social/apps/show.\$format	To provide rest services for the application registry gadget: shows application list

- **API:**

Name	Service URL Endpoint	Description
showApps	\$restContextName/social/apps/show.\$format	To show applications by the json/xml format.

Parameter	Expected values
format	xml/json

Example:

<http://localhost:9090/rest/apps/show.json>

3.4.1.3. Identity REST service

Name	Service URL	Description
IdentityRestService	\$portalName/social/identity/\$usernameToId	To get identityId by the username.

- **API:**

Name	Service URL Endpoint	Description	Example
UserId getId	\$portalName/social/identity/To get the user by username and return by the json format.	To get the user by username and return by the json format.	

Example:

<http://localhost:9090/Socialdemo/social/identity/John/show.json>

3.4.1.4. Linkshare REST service

Name	Service URL	Description
LinkshareRestService	\$restContextName/social/linkshare	To get information from a provided link.

- **API:**

Name	Service URL Endpoint	Description
getLink	\$restContextName/social/linkshare/show.\$format	To get the link content by posting with linkShare request as post data

Parameter	Expected values
format	xml/json

Example:

<http://localhost:9090/rest/social/linkshare/show.json>

3.4.1.5. Spaces REST service

Name	Service URL	Description
SpacesRestService	\$portalName/social/spaces	To provide rest services for space gadget to display user's spaces and pending spaces

- **API:**

Name	Service URL Endpoint	Description
showMySpaceList	\$restContextName/social/spaces/\$userId/mySpaces/\$spaceId	To get the user's spaces by the json/xml format.

Name	Service URL Endpoint	Description
showPendingSpaceList	<code>\$restContextName/social/spaces/\$userId/showPendingSpaceList</code>	Retrieve the pending spaces of a user. The response is in json/xml format.

Parameter	Expected values
format	xml/json

Example:

<http://localhost:8080/rest/social/spaces/s08d397dg6/mySpaces/show.xml>

3.4.2. Location

- **Maven groupId:** org.exoplatform.social
- **ArtifactId:** exo.social.component.service

3.5. Public Javascript APIs

All references for Opensocial Javascript APIs can be viewed [here](#)