

## 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 .....              | 2  |
| 1.2.1. Activity Stream .....                      | 2  |
| 1.2.2. Social RSS Reader .....                    | 3  |
| 1.2.3. My Connections .....                       | 3  |
| 1.2.4. My Spaces .....                            | 3  |
| 2. Configuration .....                            | 4  |
| 2.1. Components .....                             | 4  |
| 2.1.1. ActivityManager .....                      | 4  |
| 2.1.2. SpaceService .....                         | 4  |
| 2.1.3. IdentityManager .....                      | 4  |
| 2.1.4. ProfileConfig .....                        | 4  |
| 2.1.5. ServiceProviderStore .....                 | 5  |
| 2.2. External component plugins .....             | 5  |
| 2.2.1. MentionsProcessor .....                    | 5  |
| 2.2.2. PortletPreferenceRequiredPlugin .....      | 5  |
| 2.2.3. SpaceApplicationConfigPlugin .....         | 5  |
| 2.2.4. AddNodeTypePlugin .....                    | 5  |
| 2.3. RelationshipManager .....                    | 6  |
| 2.4. SpaceIdentityProvider .....                  | 6  |
| 2.5. SpaceApplicationHandler .....                | 6  |
| 2.6. ExoPeopleService .....                       | 6  |
| 2.7. Space Service .....                          | 6  |
| 2.7.1. Description .....                          | 6  |
| 2.7.2. Components configuration .....             | 6  |
| 2.7.3. External plug-in configuration .....       | 7  |
| 2.8. Activity Manager .....                       | 9  |
| 2.8.1. Description .....                          | 9  |
| 2.8.2. Component plug-in configuration .....      | 9  |
| 2.8.3. External plug-in configuration .....       | 10 |
| 2.9. Identity Manager .....                       | 10 |
| 2.9.1. Description .....                          | 10 |
| 2.9.2. Component plug-in configuration .....      | 10 |
| 2.10. OpenSocial Rest Context Configuration ..... | 11 |
| 2.10.1. Description .....                         | 11 |
| 2.10.2. Component plug-in configuration .....     | 11 |
| 2.11. Spaces Template configuration .....         | 12 |
| 2.12. Configure the Oauth 2 legged scenario ..... | 13 |
| 2.12.1. Generate the certificates .....           | 13 |
| 2.12.2. Configure the property file .....         | 14 |
| 3. Developers References .....                    | 15 |
| 3.1. UI Extensions .....                          | 15 |
| 3.1.1. About Activity Plug-in .....               | 15 |
| 3.1.2. Create the activity plug-in .....          | 15 |
| 3.2. Overridable Components .....                 | 19 |
| 3.3. Public Java APIs .....                       | 20 |
| 3.3.1. ActivityManager .....                      | 20 |
| 3.3.2. IdentityManager .....                      | 22 |
| 3.3.3. RelationshipManager .....                  | 25 |
| 3.3.4. SpaceService .....                         | 27 |
| 3.4. Public REST APIs .....                       | 33 |

---

|                                      |    |
|--------------------------------------|----|
| 3.4.1. Activities REST service ..... | 33 |
| 3.4.2. Apps REST service .....       | 35 |
| 3.4.3. Identity REST service .....   | 35 |
| 3.4.4. Linkshare REST service .....  | 36 |
| 3.4.5. People REST Service .....     | 36 |
| 3.5. Sample code/tutorial .....      | 39 |
| 3.5.1. Activity Stream .....         | 39 |
| 3.5.2. People .....                  | 43 |
| 3.5.3. Spaces .....                  | 46 |
| 3.5.4. Open Social .....             | 48 |
| 3.5.5. Space widget tutorial .....   | 50 |
| 3.6. Public Javascript APIs .....    | 51 |

---

# Chapter 1. Applications

## 1.1. List of Portlets in Social

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

| Portlets name                  | Description                                                                                                                                                                                      |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SpaceMenuPortlet               | Display the menu on the right of a Space: Home, Dashboard, Members, SpaceSettings                                                                                                                |
| MembersPortlet                 | Display MemberSpace where you can search for space members or list space members in alphabet                                                                                                     |
| SpaceSettingPortlet            | Shows the settings of Space: Settings, Visibility, Members, Application. You can change the setting information of a space if you are the creator of that space or have the manager right on it. |
| MySpacesPortlet                | Displays your Space page where you can add new spaces, search for spaces or list spaces in alphabet                                                                                              |
| InvitationSpacesPortlet        | Displays the <b>Space invitation</b> page where you can search for spaces, list spaces in alphabet and view all the Space invitations                                                            |
| PendingSpacesPortlet           | Displays the <b>Space requests to join</b> page where you can search for spaces, list spaces in alphabet and view all the pending request to join Spaces                                         |
| PublicSpaces Portlet           | Displays the <b>Public Spaces</b> page where you can search for spaces, list spaces by alphabet and view all available public spaces to join                                                     |
| SpaceActivityStreamPortlet     | Display spaces activities                                                                                                                                                                        |
| UserActivityStreamPortlet      | Update user's activities/status                                                                                                                                                                  |
| PeoplePortlet                  | The People page where you can find people, display people name by alphabet                                                                                                                       |
| ProfileNavigationPortlet       |                                                                                                                                                                                                  |
| ConnectionsPortlet             | The Conections page where you can access to your network, invitations to connect or connection requests                                                                                          |
| InvitationsPortlet             |                                                                                                                                                                                                  |
| RequestsPortlet                |                                                                                                                                                                                                  |
| UserProfilePortlet             | Display user profile page where you can view/ edit user's basic information, contacts, experience or change avatar                                                                               |
| MyConnectionsNavigationPortlet |                                                                                                                                                                                                  |

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

| Portlet name                      | Description                                                                                                                                                            |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MembersPortlet</b>             | To enable users to search for Space members or list Space members in the alphabet order.                                                                               |
| <b>MySpacesPortlet</b>            | To display the Space page where you can add new Spaces, search for Spaces or list Spaces in the alphabet order.                                                        |
| <b>SpaceActivityStreamPortlet</b> | To share Spaces activity information.                                                                                                                                  |
| <b>InvitationsPortlet</b>         | To list all people that invite users.                                                                                                                                  |
| <b>RequestsPortlet</b>            | To list all invitations that the user requests.                                                                                                                        |
| <b>InvitationSpaces Portlet</b>   | To enable users to search for Spaces, and list spaces in the alphabet order and view all the Space invitations.                                                        |
| <b>PendingSpacePortlet</b>        | To display the <b>Space requests to join</b> page where you can search for Spaces, list spaces in the alphabet order and view all the pending requests to join Spaces. |
| <b>PublicSpacesPortlet</b>        | To display the <b>Public Spaces</b> page where you can search for Spaces, list Spaces in the alphabet order and view all available public Spaces to join.              |
| <b>UserActivityStreamPortlet</b>  | To update and share user's activities and/or status.                                                                                                                   |
| <b>People Portlet</b>             | To display the People page where you can find people, display people names in the alphabet order.                                                                      |
| <b>ConnectionsPortlet</b>         | To display the Connections page where you can access your network, invitations to connect or connection requests.                                                      |
| <b>UserProfilePortlet</b>         | 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 *opensocial.war*.

### 1.2.1. Activity Stream

- **Description:** Manages activities of users. Some kinds of activities such as: update status, like/unlike activities, comment activities, delete activities and delete comments.
- **Used REST services:** ActivitiesRestServices

### 1.2.2. Social RSS Reader

- **Description:** Fetches and parses RSS information from a specific url.
- **Description of user preferences:** The number of RSS per page is 10 by default. Get RSS from input URL (Default URL is <http://blog.exoplatform.org/feed/>).
- **Used REST services:** N/A

### 1.2.3. My Connections

- **Description:** Displays a viewer's information and his connections' information.
- **Description of user preferences:** Connections displayed per page. The number of connections displayed is 5 by default.
- **Used REST services:** N/A

### 1.2.4. My Spaces

- **Description:** Displays all the spaces that a user has the "member" role.
- **Used REST services:** SpacesRestService

---

## Chapter 2. Configuration

### 2.1. Components

#### 2.1.1. ActivityManager

| Configuration name | Data type   | Default Value                                     | Description |
|--------------------|-------------|---------------------------------------------------|-------------|
| allowedTags        | String list | b, i, a, span, em, strong, p, ol, ul, li, br, img | html tags   |

#### 2.1.2. SpaceService

| Configuration name | Data type   | Default Value                                                         | Description                      |
|--------------------|-------------|-----------------------------------------------------------------------|----------------------------------|
| space.homeNodeApp  | String      | SpaceActivityStreamPortlet                                            | The home application for a space |
| space.apps         | String list | DashboardPortlet:true, SpaceSettingPortlet:false, MembersPortlet:true | The space applications           |



#### Note

Deprecated: `org.exoplatform.social.core.space.SpaceApplicationConfigPlugin` Use `external-component-plugins` instead:

#### 2.1.3. IdentityManager

| Configuration name | Data type | Default Value                                               | Description |
|--------------------|-----------|-------------------------------------------------------------|-------------|
| providers          | String    | org.exoplatform.social.core.identity.provider.SpaceIdentity |             |

#### 2.1.4. ProfileConfig

| Configuration name | Data type | Default Value       | Description |
|--------------------|-----------|---------------------|-------------|
| nodetype.emails    | String    | exo:profileKeyValue |             |
| nodetype.phones    | String    | exo:profileKeyValue |             |
| nodetype.ims       | String    | exo:profileKeyValue |             |
| nodetype.urls      | String    | exo:profileKeyValue |             |
| nodetype.address   | String    | exo:profileAddress  |             |

| Configuration name   | Data type | Default Value         | Description |
|----------------------|-----------|-----------------------|-------------|
| nodetype.experiences | String    | exo:profileExperience |             |
| nodetype.education   | String    | exo:profileEducation  |             |
| forceMultiValue      | String    | xxxxxxxxxxxxx         |             |

### 2.1.5. ServiceProviderStore

| Configuration name | Data type         | Default Value | Description             |
|--------------------|-------------------|---------------|-------------------------|
| sample-provider    | properties-params |               | sample service provider |

## 2.2. External component plugins

### 2.2.1. MentionsProcessor

| Configuration name | Data type | Default Value | Description                                           |
|--------------------|-----------|---------------|-------------------------------------------------------|
| priority           | String    | 2             | priority of this processor (lower are executed first) |

### 2.2.2. PortletPreferenceRequiredPlugin

| Configuration name    | Data type   | Default Value                                                   | Description                                          |
|-----------------------|-------------|-----------------------------------------------------------------|------------------------------------------------------|
| portletsPrefsRequired | String list | SpaceActivityStreamPortlet, SpaceSettingPortlet, MembersPortlet | The portlet name which requires space URL preference |

### 2.2.3. SpaceApplicationConfigPlugin

| Configuration name   | Data type              | Description                                   |  |
|----------------------|------------------------|-----------------------------------------------|--|
| spaceHomeApplication | SpaceApplicationConfig | The space application for the space home node |  |
| spaceApplicationList | SpaceApplicationConfig | space application list configuration          |  |

### 2.2.4. AddNodeTypePlugin

| Configuration name         | Data type | Default Value                        | Description                  |
|----------------------------|-----------|--------------------------------------|------------------------------|
| autoCreatedInNewRepository | String    | jar:/conf/portal/core-node-types.xml | Node type configuration file |

## 2.3. RelationshipManager

The Service is used to manipulate user relationships.

## 2.4. SpaceIdentityProvider

The provider is to give identity space instances.

## 2.5. SpaceApplicationHandler

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

## 2.6. ExoPeopleService

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

## 2.7. Space Service

### 2.7.1. Description

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

### 2.7.2. 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>
  <!-- Deprecated, Use external-component-plugins instead
  <init-params>
    <values-param>
      <name>space.homeNodeApp</name>
      <value>SpaceActivityStreamPortlet</value>
    </values-param>
    <values-param>
      <name>space.apps</name>
      <value>DashboardPortlet:true</value>
      <value>SpaceSettingPortlet:false</value>
      <value>MembersPortlet:true</value>
    </values-param>
  </init-params>
```

```

</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<br>SpaceSettingPortlet:true<br>MembersPortlet:true | The list of configurations for portlets used as portlet applications. |

## 2.7.3. External plug-in configuration

### 2.7.3.1. PortletPreferenceRequiredPlugin

```

<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;<br>SpaceSettingPortlet;<br>MembersPortlet | List of portlets which need to be saved and get the space context name. |

## 2.7.3.2. SpaceApplicationConfigPlugin



## Note

Since 1.1.3

```
<external-component-plugins>
  <target-component>org.exoplatform.social.core.space.spi.SpaceService</target-component>
  <!-- Default applications to be installed when creating a new space -->
  <component-plugin>
    <name>Space Application Configuration</name>
    <set-method>setSpaceApplicationConfigPlugin</set-method>
    <type>org.exoplatform.social.core.space.SpaceApplicationConfigPlugin</type>
    <init-params>

      <object-param>
        <name>spaceHomeApplication</name>
        <description>Space Home Application</description>
        <object type="org.exoplatform.social.core.space.SpaceApplicationConfigPlugin$SpaceApplication">
          <field name="portletApp"><string>social-portlet</string></field>
          <field name="portletName"><string>SpaceActivityStreamPortlet</string></field>
          <field name="appTitle"><string>Home</string></field>
          <!--<field name="icon"><string>SpaceHomeIcon</string></field>-->
        </object>
      </object-param>

      <object-param>
        <name>spaceApplicationListConfig</name>
        <description>space application list configuration</description>
        <object type="org.exoplatform.social.core.space.SpaceApplicationConfigPlugin">
          <field name="spaceApplicationList">
            <collection type="java.util.ArrayList">
              <value>
                <object type="org.exoplatform.social.core.space.SpaceApplicationConfigPlugin$SpaceApplication">
                  <field name="portletApp"><string>dashboard</string></field>
                  <field name="portletName"><string>DashboardPortlet</string></field>
                  <field name="appTitle"><string>Dashboard</string></field>
                  <field name="removable"><boolean>true</boolean></field>
                  <field name="order"><int>1</int></field>
                  <field name="uri"><string>dashboard</string></field>
                  <!--<field name="icon"><string>SpaceDashboardIcon</string></field>-->
                </object>
              </value>
              <value>
                <object type="org.exoplatform.social.core.space.SpaceApplicationConfigPlugin$SpaceApplication">
                  <field name="portletApp"><string>social-portlet</string></field>
                  <field name="portletName"><string>SpaceSettingPortlet</string></field>
                  <field name="appTitle"><string>Space Settings</string></field>
                  <field name="removable"><boolean>false</boolean></field>
                  <field name="order"><int>2</int></field>
                  <field name="uri"><string>settings</string></field>
                  <!--<field name="icon"><string>SpaceSettingsIcon</string></field>-->
                </object>
              </value>
              <value>
                <object type="org.exoplatform.social.core.space.SpaceApplicationConfigPlugin$SpaceApplication">
                  <field name="portletApp"><string>social-portlet</string></field>
                  <field name="portletName"><string>MembersPortlet</string></field>
                  <field name="appTitle"><string>Members</string></field>
                  <field name="removable"><boolean>true</boolean></field>
                  <field name="order"><int>3</int></field>
                  <field name="uri"><string>members</string></field>
                  <!--<field name="icon"><string>SpaceMembersIcon</string></field>-->
                </object>
              </value>
            </collection>
          </field>
        </object>
      </object-param>

    </init-params>
  </component-plugin>
</external-component-plugins>
```

In which:

| Name                                   | Set-method                       | Type                                             | Description                                                                         |
|----------------------------------------|----------------------------------|--------------------------------------------------|-------------------------------------------------------------------------------------|
| <b>Space Application Configuration</b> | setSpaceApplicationConfiguration | org.exoplatform.social.core.configuration.Plugin | Configuration for the space applications to be installed when creating a new space. |

## 2.8. Activity Manager

### 2.8.1. Description

The service is to manipulate space and user activities.

### 2.8.2. Component plug-in configuration

```
<component-plugin>
  <name>OSHtmlSanitizer</name>
  <set-method>addProcessorPlugin</set-method>
  <type>org.exoplatform.social.core.processor.OSHtmlSanitizerProcessor</type>
  <init-params>
    <values-param>
      <name>allowedTags</name>
      <value>b</value>
      <value>i</value>
      <value>a</value>
      <value>span</value>
      <value>em</value>
      <value>strong</value>
      <value>p</value>
      <value>ol</value>
      <value>ul</value>
      <value>li</value>
      <value>br</value>
      <value>img</value>
    </values-param>
  </init-params>
</component-plugin>
```

In which,

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

- Init-params:

| Name               | Possible value | Default Value                                     | Description                                                     |
|--------------------|----------------|---------------------------------------------------|-----------------------------------------------------------------|
| <b>allowedTags</b> | 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). |

### 2.8.3. 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                                                                             |
|--------------------------|--------------------|---------------------------------------------------------|-----------------------------------------------------------------------------------------|
| <b>MentionsProcessor</b> | addProcessorPlugin | org.exoplatform.social.core.processor.MentionsProcessor | Processes mentions and substitutes @username expressions by a link on the user profile. |

- Init-params:

| Name            | Possible value  | Default Value | Description                                                          |
|-----------------|-----------------|---------------|----------------------------------------------------------------------|
| <b>priority</b> | priority number | 2             | Priority of this processor (The lower level will be executed first). |

## 2.9. Identity Manager

### 2.9.1. Description

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

### 2.9.2. 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                                   |
|-------------------------------------|---------------------------|-------------------------------------------------------------|-----------------------------------------------|
| <b>SpaceIdentityProvider plugin</b> | registerIdentityProviders | org.exoplatform.social.core.identity.IdentityProviderPlugin | Identity Provider plugin identity for a space |

- Init-params:

| Name             | Possible value                 | Default Value                                                  | Description                                          |
|------------------|--------------------------------|----------------------------------------------------------------|------------------------------------------------------|
| <b>providers</b> | Every other identity providers | org.exoplatform.social.identity.provider.SpaceIdentityProvider | Identity Provider instances for managing identities. |

## 2.10. OpenSocial Rest Context Configuration

### 2.10.1. Description

The service is used to configure the portal container name when there is a OpenSocial REST API request. By configuring this service, we can make sure to reach the right portal container. The default portal container is portal.

### 2.10.2. Component plug-in configuration

This should be used when there is a portal container different than the default portal one.

```

<external-component-plugins>
  <target-component>org.exoplatform.social.opensocial.auth.RestPortalContainerNameConfig</target-component>
  <component-plugin>
    <name>set portal container name used for REST service</name>
    <set-method>setRestContainerName</set-method>
    <type>org.exoplatform.social.opensocial.auth.RestPortalContainerNameConfig</type>
    <init-params>
      <value-param>
        <name>rest-container-name</name>
        <value>socialdemo</value>
      </value-param>
    </init-params>
  </component-plugin>
</external-component-plugins>

```

```
</component-plugin>
</external-component-plugins>
```

In which:

| Name                                                   | Set-method           | Type                                                              | Description                                                                             |
|--------------------------------------------------------|----------------------|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| <b>set portal container name used for REST service</b> | setRestContainerName | org.exoplatform.social.openSocialPluginRestProviderContainerNameC | The plugin Rest Provider set rest portal container name for OpenSocial REST API request |

- Init-params:

| Name                       | Possible value                  | Default Value | Description           |
|----------------------------|---------------------------------|---------------|-----------------------|
| <b>rest-container-name</b> | any valid portal container name | N/A           | Portal container name |

## 2.11. Spaces Template configuration

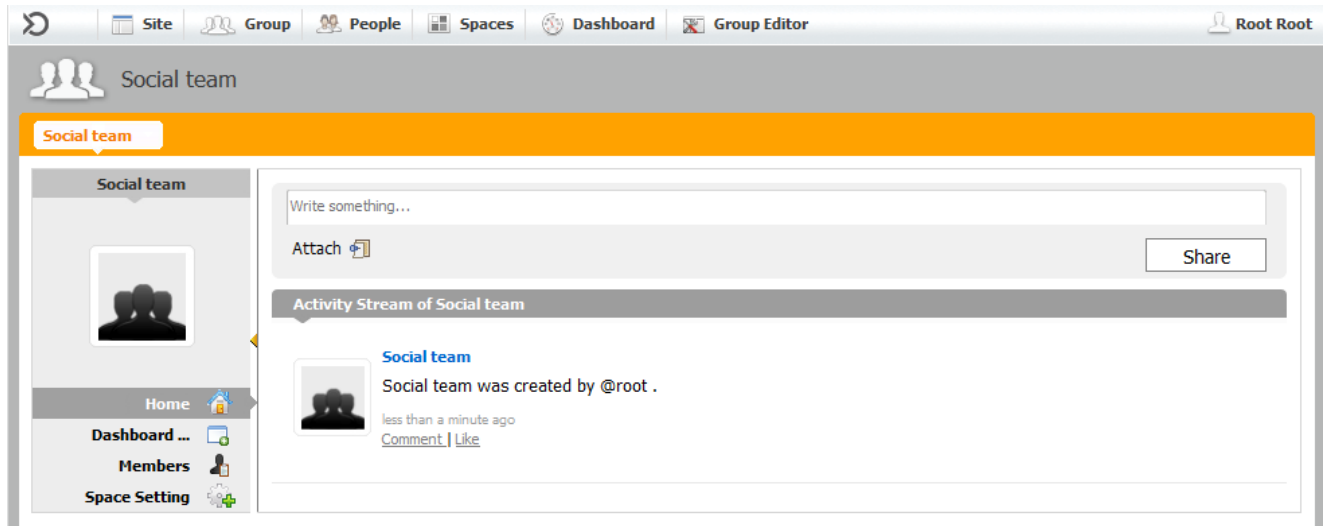
In the eXo Spaces, we may have two space types (classic and webos spaces). This is for the classic mode (it's the only one implemented now).

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">
      <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>
  </container>
</page>
```

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

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

## 2.12. 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](#)

### 2.12.1. 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
```

## 2.12.2. 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

Choose archetype:
Choose a number: 76
Choose version: 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.

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

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

```

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

  <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;

@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;

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-instance">
  <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.SimpleSpaceUIActivityBuilder</string></field>
          </object>
        </object-param>
      </init-params>
    </component-plugin>
  </external-component-plugins>
</configuration>

```

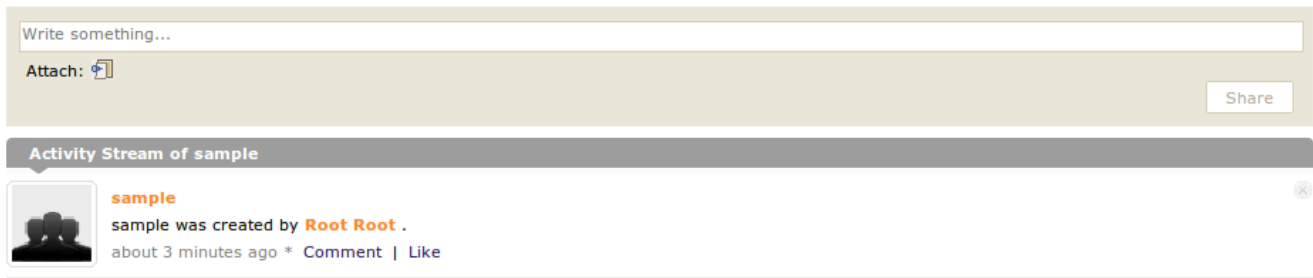
Note that `exosocial:spaces` is defined in

```
<field name="name"><string>exosocial:spaces</string></field>
```

, its value must match the activity's type you want to display with your UI component.

Build sample project and copy the jar 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":



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;

@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_Delete_Activity"),
        @EventConfig(listeners = BaseUIActivity.DeleteCommentActionListener.class, confirm = "UIActivity.msg.Are_You_Sure_Delete_Comment")
    }
)
public class UISpaceLookAndFeelActivity extends BaseUIActivity {
}
```

Now, create the `UISpaceLookAndFeelActivity` template by copying the content of [UIDefaultActivity.gtmpl](#) to this template file.

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

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

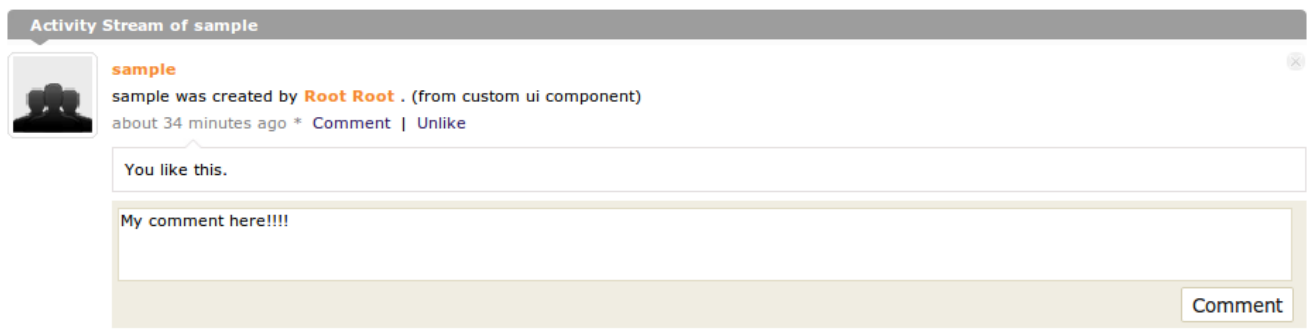
We need to reconfigure `configuration.xml`:

```

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



## 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.

### 3.1.2.1.1. What is activity builder?



## Note

to do

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



## Note

to do

## 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. Public Java APIs

### 3.3.1. ActivityManager

| Method                                                                         | Param                                                                                 | Return         | Description                                                                                                                     |
|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>saveActivity</b> (Identity owner, Activity activity)                        | owner - the owner of activity stream, activity - the activity which needs to be saved | Activity       | Saves an activity to the stream of a owner. Note that the Activity.userId will be set to the owner identity if not already set. |
| <b>getActivity</b> (java.lang.String activityId)                               | activityId - the id of activity                                                       | Activity       | Gets an activity by activity id.                                                                                                |
| <b>deleteActivity</b> (java.lang.String activityId)                            | activityId - the id of activity                                                       | void           | Deletes an activity by id.                                                                                                      |
| <b>deleteComment</b> (java.lang.String activityId, java.lang.String commentId) | activityId - the id of activity, commentId - the id of comment                        | void           | Deletes comment by id.                                                                                                          |
| <b>getActivities</b> (Identity identity)                                       | identity                                                                              | List<Activity> | Gets the latest activities by identity with the default limit of 20 latest activities.                                          |
| <b>getActivities</b> (Identity identity, start - the                           | identity, start - the                                                                 | List<Activity> | Gets the latest activities                                                                                                      |

| Method                                                                                       | Param                                                              | Return         | Description                                                                                                                                                                                                                                                   |
|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| identity, long start, long limit)                                                            | offset index , limit                                               |                | by identity, specifying the start offset index and limit.                                                                                                                                                                                                     |
| <b>getActivitiesOfConnections</b> (Identity identity, ownerId)                               | identity                                                           | List<Activity> | Gets activities of connections from an identity. The activities are sorted by time. The result list is maximum 30 activities by default. (Since 1.1.1).                                                                                                       |
| <b>getActivitiesOfConnections</b> (Identity identity, limit, ownerId, int offset, int limit) | identity, limit, offset                                            | List<Activity> | Gets activities of connections from an identity by specifying offset and limit. The activities are returned as a list that is sorted starting from the most recent activity. (Since 1.1.3).                                                                   |
| <b>getActivitiesOfUserSpaces</b> (Identity identity, ownerId)                                | identity                                                           | List<Activity> | Gets the activities from all user's spaces.                                                                                                                                                                                                                   |
| <b>getActivityFeed</b> (Identity identity)                                                   | identity                                                           | List<Activity> | Gets the activity feed of an identity. This feed is the combination of all the activities of his own activities, his connections' activities and his spaces' activities which are sorted by time. The latest activity is the first item in the activity list. |
| <b>saveActivity</b> (Activity activity)                                                      | activity - the activity which needs to be saved                    | activity       | Saves the activity into the stream for the activity's userId.                                                                                                                                                                                                 |
| <b>saveComment</b> (Activity activity, Activity comment)                                     | activity, comment                                                  | void           | Saves new or update comments to an activity comment which is an instance of Activity with mandatory properties: userId, title.                                                                                                                                |
| <b>saveLike</b> (Activity activity, Identity identity)                                       | activity, identity                                                 | void           | Saves an identity of user who likes the activity.                                                                                                                                                                                                             |
| <b>removeLike</b> (Activity activity, Identity identity)                                     | activity, identity - the identity of user who unlikes the activity | void           | Removes the like activity, if this activity liked, remove, else does nothing.                                                                                                                                                                                 |

| Method                                                                                                       | Param                                                                                                          | Return          | Description                                      |
|--------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|-----------------|--------------------------------------------------|
| <b>getComments</b> (Activity activity)                                                                       | activity                                                                                                       | comments        | Gets an activity's commentList.                  |
| <b>recordActivity</b> (Identity owner, java.lang.String type, java.lang.String title, java.lang.String body) | owner - the owner of the target stream for this activity, type is the type of activity (freeform), title, body | stored activity | Records an activity.                             |
| <b>recordActivity</b> (Identity owner, Activity activity)                                                    | owner, activity                                                                                                | stored activity | Saves an activity.                               |
| <b>addProcessor</b> (ActivityProcessor processor, Adds a new processor)                                      | processor, addProcessorPlugin                                                                                  | void            | Adds a new activity processor.                   |
| <b>addProcessorPlugin</b> (BaseActivityProcessorPlugin plugin)                                               | plugin                                                                                                         | void            | Adds a new processor plug-in.                    |
| <b>getActivitiesCount</b> (Identity owner)                                                                   | owner                                                                                                          | int             | Gets the number of activity from a stream owner. |

### 3.3.2. IdentityManager

| Method                                                           | Param                                                                                                                                                       | Return                                           | Description                                                              |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------------|
| <b>registerIdentityProviders</b> (IdentityProviderPlugin plugin) | plugin                                                                                                                                                      |                                                  | Registers one or more IdentityProvider through an IdentityProviderPlugin |
| <b>getIdentity</b> (String id)                                   | id can be a social GlobalId or a raw identity such as in Identity.getId()                                                                                   | null if nothing is found, or the Identity object | Gets the identity by id and to load his profile.                         |
| <b>getIdentity</b> (String id, boolean loadProfile)              | id can be a social GlobalId or a raw identity such as in Identity.getId(), loadProfile - the value is true if the profile is loaded and false if not loaded | null if nothing is found, or the Identity object | Gets the identity by loading id of the profile optionally.               |
| <b>deleteIdentity</b> (Identity identity)                        | identity                                                                                                                                                    | void                                             | Deletes an identity.                                                     |
| <b>addIdentityProvider</b> (IdentityProvider idProvider)         | idProvider - is the id of provider                                                                                                                          | void                                             | Adds the id of provider.                                                 |
| <b>getIdentitiesCount</b> (String providerId)                    | providerId                                                                                                                                                  | long                                             | Gets the number of identities.                                           |

| Method                                                                                                                                   | Param                                                                                                                                                                                           | Return                                                                                                               | Description                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>getOrCreateIdentity</b> (java.lang.String providerId, java.lang.String remoteId)                                                      | String is the id of provider, remoteId is the remote id                                                                                                                                         | identity                                                                                                             | Gets the identity by a remote id.                                                                                                                                                                                                                                          |
| <b>getOrCreateIdentity</b> (java.lang.String providerId, java.lang.String remoteId, boolean loadProfile)                                 | String - referring to the name of the Identity provider, remoteId - the identifier that identify the identity in the specific identity provider, loadProfile is true when the profile is loaded | null if nothing is found, or the Identity object TODO improves the performance by specifying what needs to be loaded | Returns an Identity object that is specific to a special type. For example, if the type is LinkedIn, the identifier will be the URL of profile or if it is a CS contact manager contact, it will be the UID of the contact. A new identity is created if it doesnot exist. |
| <b>getIdentitiesByProfileFilter</b> (java.lang.String providerId, ProfileFilter profileFilter) throws Exception                          | String is the id of provider, profileFilter is the filter of provider                                                                                                                           | the identities by the profile filter                                                                                 | Gets the identities by profile filter.                                                                                                                                                                                                                                     |
| <b>getIdentitiesByProfileFilter</b> (java.lang.String providerId, ProfileFilter profileFilter, long offset, long limit) throws Exception | String profileFilter, offset, limit,                                                                                                                                                            | List<Identity>                                                                                                       | Gets the identities by profile filter.                                                                                                                                                                                                                                     |
| <b>getIdentitiesByProfileFilter</b> (ProfileFilter profileFilter) throws Exception                                                       | ProfileFilter - the profile filter                                                                                                                                                              | the identities by profile filter                                                                                     | Gets the identities by profile filter.                                                                                                                                                                                                                                     |
| <b>getIdentitiesByProfileFilter</b> (ProfileFilter, profileFilter, long offset, long limit) throws Exception                             | ProfileFilter, offset, limit                                                                                                                                                                    | List<Identity>                                                                                                       | Gets the identities by profile filter.                                                                                                                                                                                                                                     |
| <b>getIdentitiesFilterByAlphaBet</b> (java.lang.String providerId, ProfileFilter profileFilter) throws Exception                         | String is the id of provider, profileFilter is the profile filter                                                                                                                               | the identities filter by alphabet                                                                                    | Gets the identities filter by alpha bet.                                                                                                                                                                                                                                   |
| <b>getIdentitiesFilterByAlphaBet</b> (String, providerId, ProfileFilter profileFilter, long offset, long limit) throws Exception         | String, profileFilter, offset, limit                                                                                                                                                            | List<Identity>                                                                                                       | Gets the identities filter by alpha bet by specifying offset and limit.                                                                                                                                                                                                    |
| <b>getIdentitiesFilterByAlphaBet</b> (ProfileFilter profileFilter) throws Exception                                                      | ProfileFilter - the profile filter                                                                                                                                                              | the identities filter by alphabet                                                                                    | Gets the identities filter by alphabet.                                                                                                                                                                                                                                    |
| <b>getIdentity</b> (java.lang.String providerId, java.lang.String remoteId, boolean loadProfile)                                         | String providerId, remoteId, loadProfile                                                                                                                                                        | Identity                                                                                                             | Gets the identity.                                                                                                                                                                                                                                                         |

| Method                                                                                                        | Param                                                                      | Return          | Description                                           |
|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|-----------------|-------------------------------------------------------|
| java.lang.String remoteId,<br>boolean loadProfile)                                                            |                                                                            |                 |                                                       |
| <b>identityExisted</b> (java.lang.String providerId, remoteId<br>providerId,<br>java.lang.String<br>remoteId) |                                                                            | boolean         | Checks if the identity is<br>already existed or not.  |
| <b>saveIdentity</b> (Identity<br>identity)                                                                    | identity is the identity                                                   | void            | Saves the identity.                                   |
| <b>saveProfile</b> (Profile<br>profile)                                                                       | profile                                                                    | void            | Saves a profile.                                      |
| <b>updateAvatar</b> (Profile p)<br>throws Exception                                                           | p - profile                                                                | void            | Updates avatar.                                       |
| <b>updateBasicInfo</b> (Profile<br>p) throws Exception                                                        | p - profile                                                                | Profile         | Updates basic info of a<br>profile.                   |
| <b>updateContactSection</b> (Profile<br>p) throws Exception                                                   | profile                                                                    | void            | Updates the contact<br>section of a profile.          |
| <b>updateExperienceSection</b> (Profile<br>p) throws Exception                                                | profile                                                                    | void            | Updates the experience<br>section of a profile.       |
| <b>updateHeaderSection</b> (Profile<br>p) throws Exception                                                    | profile                                                                    | void            | Updates the header<br>section of profile.             |
| <b>getIdentities</b> (java.lang.String<br>providerId) throws<br>Exception                                     | providerId is the id of<br>provider                                        | List<Identity>  | Gets the identity.                                    |
| <b>getIdentities</b> (java.lang.String<br>providerId, boolean<br>loadProfile)                                 | providerId is the id of<br>provider, loadProfile is<br>the loaded profile. | List<Identity>  | Gets the identities.                                  |
| <b>getConnections</b> (Identity<br>ownerIdentity) throws<br>Exception                                         | ownerIdentity                                                              | List<Identity>  | Gets connections of an<br>identity                    |
| <b>getStorage</b> ()                                                                                          | N/A                                                                        | IdentityStorage | Gets the storage.                                     |
| <b>setIdentityStorage</b> (IdentityStorage<br>identityStorage)                                                | IdentityStorage                                                            | void            | Sets identityStorage.                                 |
| <b>registerProfileListener</b> (ProfileListener<br>listener)unregisterProfileListener                         | ProfileListener                                                            | void            | Registers a profile<br>listener.                      |
| <b>unregisterProfileListener</b> (ProfileListener<br>listener)                                                | ProfileListener                                                            | void            | Unregisters a profile<br>listener.                    |
| <b>addProfileListener</b> (ProfileListener<br>plugin)                                                         | ProfileListenerPlugin                                                      | void            | Registers a profile<br>listener component<br>plug-in. |
| <b>getIdentityStorage</b> ()                                                                                  | N/A                                                                        | IdentityStorage | Gets identity storage.                                |

### 3.3.3. RelationshipManager

| Method                                                                                                                | Param                                                                           | Return             | Description                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>getRelationshipById</b> (String id) throws Exception                                                               |                                                                                 | Relationship       | Gets relationship the by id.                                                                                                                                                                                 |
| <b>invite</b> (Identity currIdentity, Identity requestedIdentity) throws Exception                                    | currIdentity - The current identity, requestedIdentity - The requested identity | Relationship       | Creates a connection invitation between two identities.                                                                                                                                                      |
| <b>confirm</b> (Relationship relationship) throws Exception                                                           | relationship                                                                    | void               | Marks a relationship as confirmed.                                                                                                                                                                           |
| <b>deny</b> (Relationship relationship) throws Exception                                                              | relationship                                                                    | void               | Denies a relationship.                                                                                                                                                                                       |
| <b>remove</b> (Relationship relationship) throws Exception                                                            | relationship                                                                    | void               | Removes a relationship.                                                                                                                                                                                      |
| <b>ignore</b> (Relationship relationship) throws Exception                                                            | relationship                                                                    | void               | Marks a relationship as ignored.                                                                                                                                                                             |
| <b>getPendingRelationships</b> (Identity identity) throws Exception                                                   |                                                                                 | List<Relationship> | Returns all the pending relationships of the identity, including sent and received relationships of the pending type.                                                                                        |
| <b>getPendingRelationships</b> (Identity identity, boolean toConfirm) throws Exception                                | Identity, toConfirm                                                             | List<Relationship> | If toConfirm is true, it returns a list of the pending relationship received, but not confirmed, if toConfirm is false, it returns a list of the relationship sent, but not confirmed yet.                   |
| <b>getPendingRelationships</b> (Identity currIdentity, List<Identity> identities, boolean toConfirm) throws Exception | Identity - the current identity, identities - the identities, toConfirm -       | List<Relationship> | Gets the pending relations in two cases:<br><br>- If toConfirm is true, it returns a list of the pending relationships received, but not confirmed.<br><br>- If toConfirm is false, it returns a list of the |

| Method                                                                                                 | Param                                                             | Return             | Description                                                                                                            |
|--------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|--------------------|------------------------------------------------------------------------------------------------------------------------|
|                                                                                                        |                                                                   |                    | relationships sent, but not confirmed yet.                                                                             |
| <b>getContacts</b> (Identity currIdentity, List<Identity> identities) throws Exception                 | currIdentity - the current identity, identities - the identities, | List<Relationship> | Gets contacts that matches the search result.                                                                          |
| <b>getContacts</b> (Identity identity) throws Exception                                                | identity                                                          | List<Relationship> | Gets the contacts of the identity.                                                                                     |
| <b>getAllRelationships</b> (Identity identity) throws Exception                                        | identity                                                          | List<Relationship> | Returns all the relationships associated with a given identity.                                                        |
| <b>getRelationshipsById</b> (String id) throws Exception                                               | id                                                                | List<Relationship> | Returns all the relationships associated with a given identityId.                                                      |
| <b>getIdentities</b> (Identity id) throws Exception                                                    | id                                                                | List<Identity>     | Returns all the identities associated with a given identity and checks whether the relation has been validated or not. |
| <b>create</b> (Identity sender, Identity receiver)                                                     | sender, receiver                                                  | Relationship       | Creates the relationship.                                                                                              |
| <b>saveRelationship</b> (Relationship relationship) throws Exception                                   | relationship                                                      | void               | Saves the relationship.                                                                                                |
| <b>updateRelationshipCached</b> (Relationship relationship)                                            | relationship                                                      | void               | Updates the relationship cached.                                                                                       |
| <b>findRoute</b> (Identity sender, Identity receiver) throws NotSupportedException                     | sender, receiver                                                  | Relationship       | Finds a route.                                                                                                         |
| <b>findRelationships</b> (Identity ownerIdentity, Relationship.Type relationshipType) throws Exception | ownerIdentity, relationshipType                                   | List<Identity>     | Finds all identities having relationshipType with the ownerIdentity.(Since 1.1.2).                                     |
| <b>getRelationshipStatus</b> (Relationship rel, Identity id)                                           | relationship                                                      | Relationship.Type  | Gets the relationship type of the relationship with the identity.                                                      |
| <b>getConnectionStatus</b> (Identity fromIdentity, Identity toIdentity) throws Exception               | fromIdentity, toIdentity                                          | Relationship.Type  | Gets the connection type between two identities. (Since 1.1.1).                                                        |

| Method                                                       | Param                      | Return | Description                         |
|--------------------------------------------------------------|----------------------------|--------|-------------------------------------|
| <b>registerListener</b> (RelationshipListener listener)      | RelationshipListener       | void   | Registers the RelationshipListener. |
| <b>unregisterListener</b> (RelationshipListener listener)    | RelationshipListener       | void   | Removes the RelationshipListener.   |
| <b>addListenerPlugin</b> (RelationshipListenerPlugin plugin) | RelationshipListenerPlugin | void   | Adds the plugin.                    |

### 3.3.4. SpaceService

| Method                                                                                     | Param                               | Return                                                                                | Description                                                                             |
|--------------------------------------------------------------------------------------------|-------------------------------------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| <b>getAllSpaces</b> () throws SpaceException                                               | N/A                                 | List<Space> - list of spaces in Social                                                | Gets all spaces in Social.                                                              |
| <b>getSpaceByName</b> (String spaceName) throws SpaceException                             | spaceName                           | Space                                                                                 | Gets a space by its space name.                                                         |
| <b>getSpacesByFirstCharacterOfName</b> (String firstCharacterOfName) throws SpaceException | firstCharacterOfName                | List<Space> - all spaces which have first character of name matched the input string. | Gets all spaces has the name starting with the input character.                         |
| <b>getSpacesBySearchCondition</b> (String condition) throws Exception                      | condition - the input condition     | List<Space> - a list of spaces                                                        | Gets all spaces which has the name or the description that matches the input condition. |
| <b>getSpaceByGroupId</b> (String groupId) throws SpaceException,                           | groupId - the group id of the space | Space                                                                                 | Gets a space by its group id.                                                           |
| <b>getSpaceById</b> (String spaceId) throws SpaceException                                 | spaceId - Id of that space          | Space - space with the id specified                                                   | Gets a space by its id.                                                                 |
| <b>getSpaceByUrl</b> (String spaceUrl) throws SpaceException                               | spaceUrl - url of space             | Space - the space with string url specified                                           | Gets a space by its url.                                                                |
| <b>getSpaces</b> (String userId) throws SpaceException                                     | userId - Id of the user             | List<Space> - all spaces of a user in which the user is a member                      | Gets spaces of a user in which that user is a member.                                   |
| <b>getAccessibleSpaces</b> (String userId) throws SpaceException                           | userId                              | List<Space> - list of spaces                                                          | Gets spaces of a user which that user has the access permission                         |
| <b>getEditableSpaces</b> (String userId) throws SpaceException                             | userId                              | List<Space> - list of spaces                                                          | Gets spaces of a user which that user has the edit permission.                          |

| Method                                                                                 | Param                                                                                              | Return                                                              | Description                                                                                                                                                                   |
|----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>getInvitedSpaces</b> (String userId) throws SpaceException                          | userId                                                                                             | List<Space> - spaces list of all user's invited spaces              | Gets a user's invited spaces and that user can accept or deny the request.                                                                                                    |
| <b>getPublicSpaces</b> (String userId) throws SpaceException                           | userId - Id of user                                                                                | List<Space> - spaces list in which the user can request to join     | Gets a user's public spaces and that user can request to join.                                                                                                                |
| <b>getPendingSpaces</b> (String userId) throws SpaceException                          | userId                                                                                             | List<Space> - spaces list in which the user can revoke that request | Gets a user's pending spaces and that the user can revoke that request.                                                                                                       |
| <b>createSpace</b> (Space space, String creator) throws SpaceException                 | space, creator                                                                                     | Space - the created space                                           | Creates a new space by creating a new group. This new group will be under <b>/Spaces</b> node. This is shorthand for calling <code>createSpace(space, creator, null)</code> . |
| <b>createSpace</b> (Space space, String creator, String groupId) throws SpaceException | space, creator, groupId - if groupId == null: create a new space by creating a new group           | Space                                                               | Creates a new space from an existing group.                                                                                                                                   |
| <b>saveSpace</b> (Space space, boolean isNew) throws SpaceException                    | space - space is saved, isNew - true if creating a new space, otherwise, update an existing space. | void                                                                | Saves a new space or updates a space.                                                                                                                                         |
| <b>deleteSpace</b> (Space space) throws SpaceException                                 | space - the space is deleted                                                                       | void                                                                | Deletes a space. When deleting a space, all of its page navigations and its group will be deleted.                                                                            |
| <b>deleteSpace</b> (String spaceId) throws SpaceException                              | spaceId                                                                                            | void                                                                | Deletes a space by its id.                                                                                                                                                    |
| <b>initApp</b> (Space space) throws SpaceException                                     | space                                                                                              | void                                                                | Initializes default applications in a space. <b>Deprecated</b> , use <b>initApps</b> (Space) instead.                                                                         |
| <b>initApps</b> (Space space) throws SpaceException                                    | space                                                                                              | void                                                                | Initializes default applications in a space. Set <b>space.homeNodeApp</b> from the configuration file to be the root page of that space node. When removing a space, make     |

| Method                                                                                   | Param                     | Return                                     | Description                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------|---------------------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                          |                           |                                            | sure to call <b>deInitApps</b> (Space) and then <b>deleteSpace</b> (Space) or <b>deleteSpace</b> (String)                                                                                                              |
| <b>deInitApps</b> (Space space) throws SpaceException                                    | space                     | void                                       | De-initializes the applications of a space. Make sure to call this method before <b>deleteSpace</b> (Space) or <b>deleteSpace</b> (String). Otherwise, the space is deleted but its pages and navigation still exists. |
| <b>addMember</b> (Space space, String userId) throws SpaceException                      |                           | void                                       | Adds a user to a space, the user will get the "member" role in a space.                                                                                                                                                |
| <b>addMember</b> (String spaceId, String userId) throws SpaceException                   | spaceId, userId           | void                                       | Adds a user to a space, the user will get the "member" role in a space.                                                                                                                                                |
| <b>removeMember</b> (Space space, String userId) throws SpaceException                   | space, userId             | void                                       | Removes a member from a space. If the member is the only leader of that space, the member removed is not allowed and throws <b>SpaceException</b> with Code = <b>USERONLYLEADER</b> .                                  |
| <b>removeMember?</b> (String spaceId, String userId) throws SpaceException               | spaceId, userId           | void                                       | Removes a member from a space.                                                                                                                                                                                         |
| <b>getMembers</b> (Space space) throws SpaceException                                    | space                     | List<String> - a list of the space members | Gets a list of the space members from a space.                                                                                                                                                                         |
| <b>getMembers</b> (String spaceId) throws SpaceException                                 | spaceId                   | List<String> - a list of the space members | Gets a list of the space members from a space.                                                                                                                                                                         |
| <b>setLeader</b> (Space space, String userId, boolean isLeader) throws SpaceException    | space, userId, isLeader   | void                                       | Sets a member of a space as a manager.                                                                                                                                                                                 |
| <b>setLeader</b> (String spaceId, String userId, boolean isLeader) throws SpaceException | spaceId, userId, isLeader | void                                       | Sets a member of a space as a manager.                                                                                                                                                                                 |

| Method                                                                           | Param            | Return                                                                        | Description                                                      |
|----------------------------------------------------------------------------------|------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------|
| <b>isLeader</b> (Space space, String userId) throws SpaceException               | space, userId    | boolean - true if that the user is a leader, otherwise, false                 | Checks whether a user is a space's leader or not.                |
| <b>isLeader</b> (String spaceId, String userId) throws SpaceException            | spaceId, userId  | boolean - true if that user is a leader, otherwise, false                     | Checks whether a user is a space's leader or not.                |
| <b>isOnlyLeader</b> (Space space, String userId) throws SpaceException           | space, userId    | boolean - true if that user is the only leader of the space, otherwise, false | Checks whether a user is the only leader of a space or not.      |
| <b>isOnlyLeader</b> (String spaceId, String userId) throws SpaceException        | spaceId, userId  | boolean                                                                       | Checks whether a user is the only leader of a space or not.      |
| <b>isMember</b> (Space space, String userId) throws SpaceException               | space, userId    | boolean - true if that user is a member, otherwise, false                     | Checks whether a user is a space's member or not.                |
| <b>isMember</b> (String spaceId, String userId) throws SpaceException            | spaceId, userId, | boolean - true if that user is a member, otherwise, false                     | Checks whether a user is a space's member or not.                |
| <b>hasAccessPermission</b> (Space space, String userId) throws SpaceException    | space, userId    | boolean - true If the user is root or the space's member.                     | Checks if a user can access a space or not.                      |
| <b>hasAccessPermission</b> (String spaceId, String userId) throws SpaceException | spaceId, userId  | boolean - true If the user is root or the space's member                      | Checks if a user can access a space or not.                      |
| <b>hasEditPermission</b> (Space space, String userId) throws SpaceException      | space, userId    | boolean - true If the user is root or the space's manager                     | Checks if a user can have the edit permission of a space or not. |
| <b>hasEditPermission</b> (String spaceId, String userId) throws SpaceException   | spaceId, userId  | boolean - true If user is root or the space's manager                         | Checks if a user can have the edit permission of a space or not. |
| <b>isInvited</b> (Space space, String userId) throws SpaceException              | space, userId    | boolean - true if that user is in the invited list, otherwise, false          | Checks if a user is in the invited list of a space or not.       |
| <b>isInvited</b> (String spaceId, String userId) throws SpaceException           | spaceId, userId  | boolean - true if the user is in the invited list, otherwise, false           | Checks if a user is in the invited list of a space or not.       |
| <b>isPending</b> (Space space, String userId) throws SpaceException              | space, userId    | boolean - true if that user is in the pending list, otherwise, false          | Checks if a user is in the pending list of a space or not.       |
| <b>installApplication</b> (String spaceId, String appId) throws SpaceException   | spaceId, appId   | void                                                                          | Installs an application to a space.                              |

| Method                                                                                        | Param                 | Return | Description                                                                              |
|-----------------------------------------------------------------------------------------------|-----------------------|--------|------------------------------------------------------------------------------------------|
| <b>installApplication</b> (Space space, String appId) throws SpaceException                   | space, appId          | void   | Installs an application to a space                                                       |
| <b>activateApplication</b> (Space space, String appId) throws SpaceException                  | space, appId          | void   | Activates an installed application in a space.                                           |
| <b>activateApplication</b> (String spaceId, String appId) throws SpaceException               | spaceId, appId        | void   | Activates an installed application in a space.                                           |
| <b>deactivateApplication</b> (Space space, String appId) throws SpaceException                | space, appId          | void   | Deactivates an installed application in a space.                                         |
| <b>deactivateApplication</b> (String spaceId, String appId) throws SpaceException             | spaceId, appId        | void   | Deactivates an installed application in a space.                                         |
| <b>removeApplication</b> (Space space, String appId, String appName) throws SpaceException    | space, appId, appName | void   | Removes an installed application from a space.                                           |
| <b>removeApplication</b> (String spaceId, String appId, String appName) throws SpaceException | space, appId, appName | void   | Removes an installed application from a space.                                           |
| <b>requestJoin</b> (Space space, String userId) throws SpaceException                         | space, userId         | void   | Requests a user to join a space, adds that user to the pending list of the space.        |
| <b>requestJoin</b> (String spaceId, String userId) throws SpaceException                      | spaceId, userId       | void   | Requests a user to join a space, adds that user to the pending list of the space.        |
| <b>revokeRequestJoin</b> (Space space, String userId) throws SpaceException                   | space, userId         | void   | Revokes a join request after users request to join a group and is in the pending status. |
| <b>revokeRequestJoin</b> (String spaceId, String userId) throws SpaceException                | spaceId, userId       | void   | Revokes a request to join a space.                                                       |
| <b>inviteMember</b> (Space space, String userId) throws SpaceException                        | space, userId         | void   | Invites a userId to become a member of a space.                                          |
| <b>inviteMember</b> (String spaceId, String userId)                                           | spaceId, userId       | void   | Invites a userId to a be member of a space.                                              |

| Method                                                                                  | Param                             | Return | Description                                                                      |
|-----------------------------------------------------------------------------------------|-----------------------------------|--------|----------------------------------------------------------------------------------|
| throws SpaceException                                                                   |                                   |        |                                                                                  |
| <b>revokeInvitation</b> (Space space, String userId)<br>throws SpaceException           | space, userid                     | void   | Revokes an invitation. Removes a user from the invited member list of the space. |
| <b>revokeInvitation</b> (String spaceId, String userId)<br>throws SpaceException        | spaceId, userid                   | void   | Revokes an invitation. Removes a user from the invited member list of the space. |
| <b>acceptInvitation</b> (Space space, String userId)<br>throws SpaceException           | space, userid                     | void   | Accepts an invitation and moves a user from the invited list to the member list. |
| <b>acceptInvitation</b> (String spaceId, String userId)<br>throws SpaceException        | spaceId, userid                   | void   | Accepts an invitation and moves a user from the invited list to the member list. |
| <b>denyInvitation</b> (Space space, String userId)<br>throws SpaceException             | space, userid                     | void   | Denies an invitation and removes a user from the invited list.                   |
| <b>denyInvitation</b> (String spaceId, String userId)<br>throws SpaceException          | spaceId, userid                   | void   | Denies an invitation and removes a user from the invited list.                   |
| <b>validateRequest</b> (Space space, String userId)<br>throws SpaceException            | space, userid                     | void   | Validates a request and moves a user from the pending list to the member list.   |
| <b>validateRequest</b> (String spaceId, String userId)<br>throws SpaceException         | spaceId, userid                   | void   | Validates a request and moves a user from the pending list to the member list.   |
| <b>declineRequest</b> (Space space, String userId)<br>throws SpaceException             | space, userid                     | void   | Declines a request and removes a user from the pending list.                     |
| <b>declineRequest</b> (String spaceId, String userId)<br>throws SpaceException          | spaceId, userid                   | void   | Declines a request and removes a user from the pending list.                     |
| <b>registerSpaceLifecycleListener</b> (SpaceLifeCycleListener listener)                 | listener (SpaceLifeCycleListener) | void   | Registers a space lifecycle listener.                                            |
| <b>unregisterSpaceLifecycleListener</b> (SpaceLifeCycleListener listener)               | listener (SpaceLifeCycleListener) | void   | Unregisters a space lifecycle listener.                                          |
| <b>setPortletsPrefsRequired</b> (PortletPrefsRequiredPlugin portletPrefsRequiredPlugin) | PortletPrefsRequiredPlugin        | void   | Sets the portlet preferences got from the                                        |

| Method                                                               | Param                                                      | Return                       | Description                                                                                        |
|----------------------------------------------------------------------|------------------------------------------------------------|------------------------------|----------------------------------------------------------------------------------------------------|
|                                                                      |                                                            |                              | plug-in configuration.                                                                             |
| <b>getPortletsPrefsRequired()</b>                                    | N/A                                                        | String[]                     | Gets the portlet preferences required to use in creating the portlet application.                  |
| <b>setSpaceApplicationConfigPlugin(spaceApplicationConfigPlugin)</b> | spaceApplicationConfigPlugin(SpaceApplicationConfigPlugin) | SpaceApplicationConfigPlugin | Sets configuration plugin for initializing applications when creating a new space. (Since 1.1.3).  |
| <b>getSpaceApplicationConfigPlugin()</b>                             |                                                            | SpaceApplicationConfigPlugin | Gets the configuration of applications to be initialized when creating a new space. (Since 1.1.3). |

## 3.4. Public REST APIs

### 3.4.1. Activities REST service

| Name                         | Service URL                    | Location                                                                                                                             | Description                                                                                      |
|------------------------------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| <b>ActivitiesRestService</b> | {restContextName}/{portalName} | Name}/social/activities<br><b>Maven groupId:</b><br>org.exoplatform.social<br><br><b>ArtifactId:</b><br>exo.social.component.service | Provides REST services for activity applications, such as like/unlike, comment, delete activity. |

- **API:**

| Name                   | Service URL                                                           | Parameters                                 | Expected Values                                 | Description                                                                   |
|------------------------|-----------------------------------------------------------------------|--------------------------------------------|-------------------------------------------------|-------------------------------------------------------------------------------|
| <b>destroyActivity</b> | {restContextName}/{portalName}/social/activities/destroy/{activityId} | portalName<br><br>activityId<br><br>format | String<br><br>String<br><br>String: json or xml | Deletes an activity and returns the json/xml format.                          |
| <b>showLikes</b>       | {restContextName}/{portalName}/social/activities/{activityId}/likes   | portalName                                 | String                                          | Shows the list of identityId who likes this activity and returns the json/xml |

| Name                  | Service Endpoint   | URL                                                              | Parameters                                       | Expected Values                                   | Description                                                                                    |
|-----------------------|--------------------|------------------------------------------------------------------|--------------------------------------------------|---------------------------------------------------|------------------------------------------------------------------------------------------------|
|                       |                    |                                                                  | activityId<br>format                             | String<br>String: json or xml                     | format.                                                                                        |
| <b>updateLike</b>     | {restContextName}/ | {portalName}/social/activities/{activityId}/likes/{identityId}   | portalName<br>activityId<br>format               | String<br>String<br>String: json or xml           | Updates the list of likes by the json/xml format.                                              |
| <b>destroyLike</b>    | {restContextName}/ | {portalName}/social/activities/{activityId}/likes/{identityId}   | portalName<br>activityId<br>identityId<br>format | String<br>String<br>String<br>String: json or xml | Destroys like by identityId and returns the json/xml format.                                   |
| <b>showComments</b>   | {restContextName}/ | {portalName}/social/activities/{activityId}/comments/{commentId} | portalName<br>activityId<br>format               | String<br>String<br>String: json or xml           | Shows the comment list of an activity by the json/xml format.                                  |
| <b>updateComment</b>  | {restContextName}/ | {portalName}/social/activities/{activityId}/comments/{commentId} | portalName<br>activityId<br>format               | String<br>String<br>String: json or xml           | Updates update. {format} comment of an acitivity by the json/xml format.                       |
| <b>destroyComment</b> | {restContextName}/ | {portalName}/social/activities/{activityId}/comments/{commentId} | portalName<br>activityId<br>commentId<br>format  | String<br>String<br>String<br>String: json or xml | Destroys/destroy/{commentId}. {format} comment of an identity and returns the json/xml format. |

Example:

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

### 3.4.2. Apps REST service

| Name                   | Service URL                   | Location                                                                                                  | Description                                                                                   |
|------------------------|-------------------------------|-----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <b>AppsRestService</b> | {restContextName}/social/apps | <b>Maven groupId:</b><br>org.exoplatform.social<br><br><b>ArtifactId:</b><br>exo.social.component.service | Provides REST services for showing the application list from the application registry gadget. |

- **API:**

| Name            | Service URL                                 | Parameters | Expected Values     | Description                                |
|-----------------|---------------------------------------------|------------|---------------------|--------------------------------------------|
| <b>showApps</b> | {restContextName}/social/apps/show.{format} |            | String: json or xml | Shows applications by the json/xml format. |

Example:

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

### 3.4.3. Identity REST service

| Name                       | Service URL                                                                    | Location                                                                                                  | Description                          |
|----------------------------|--------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|--------------------------------------|
| <b>IdentityRestService</b> | {restContextName}/{portalName}/social/identity/{username}/id/show/{identityId} | <b>Maven groupId:</b><br>org.exoplatform.social<br><br><b>ArtifactId:</b><br>exo.social.component.service | Gets the identityId by the username. |

- **API:**

| Name         | Service URL                                                       | Parameters                 | Expected Values      | Description                                                   |
|--------------|-------------------------------------------------------------------|----------------------------|----------------------|---------------------------------------------------------------|
| <b>getId</b> | {restContextName}/{portalName}/social/identity/{username}/id/show | username<br><br>portalname | String<br><br>String | Gets the identity by username and returns by the json format. |

| Name | Service URL | Parameters | Expected Values | Description |
|------|-------------|------------|-----------------|-------------|
|      |             |            |                 |             |

Example:

<http://localhost:8080/rest-socialdemo/socialdemo/social/identity/john/id/show.json>

### 3.4.4. Linkshare REST service

| Name                        | Service URL               | Location                                                                                                               | Description                                                                                                               |
|-----------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>LinkshareRestService</b> | {restContextName}/social/ | linkshare<br><b>Maven groupId:</b><br>org.exoplatform.social<br><br><b>ArtifactId:</b><br>exo.social.component.service | Provides REST service for getting metadata information, such as page title, images, and description from a provided link. |

• API:

| Name           | Service URL                                      | Parameters | Expected Values      | Description                                                                  |
|----------------|--------------------------------------------------|------------|----------------------|------------------------------------------------------------------------------|
| <b>getLink</b> | {restContextName}/social/linkshare/show.{format} | format     | {format} json or xml | Gets the provided link metadata by posting a linkShare request as post data. |

Example:

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

### 3.4.5. People REST Service

| Name                     | Service URL               | Location                                                                                                            | Description                                                                                                             |
|--------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>PeopleRestService</b> | {restContextName}/social/ | people<br><b>Maven groupId:</b><br>org.exoplatform.social<br><br><b>ArtifactId:</b><br>exo.social.component.service | Provides REST services for serving requests related to people, such as suggesting username, getting people information. |

• API:

| Name                    | Service URL                                                                                         | Parameters          | Expected Values     | Description                                                                                    |
|-------------------------|-----------------------------------------------------------------------------------------------------|---------------------|---------------------|------------------------------------------------------------------------------------------------|
| <b>suggestUsernames</b> | {restContextName}/social/people/suggest.                                                            | nameToSearch        | {format} String     | Gets and returns a list of users's full names that matches the input string for suggesting.    |
|                         |                                                                                                     | currentUser         | String              |                                                                                                |
|                         |                                                                                                     | typeOfRelation      | String              |                                                                                                |
|                         |                                                                                                     | spaceURL            | String              |                                                                                                |
|                         |                                                                                                     | portalName          | String              |                                                                                                |
|                         |                                                                                                     | format              | String              |                                                                                                |
| <b>getPeopleInfo</b>    | {restContextName}/social/people/{portalName}/{currentUserUserName}/getPeopleInfo/{userId}. {format} | portalName          | String              | Gets people information of people that is displayed as users's detailed information on pop-up. |
|                         |                                                                                                     | currentUserUserName | String              |                                                                                                |
|                         |                                                                                                     | userId              | String              |                                                                                                |
|                         |                                                                                                     | format              | String: json or xml |                                                                                                |
|                         |                                                                                                     | updatedType         | String              |                                                                                                |

Example:

<http://localhost:8080/rest-socialdemo/social/people/suggest.json?nameToSearch=root&currentUser=john&typeOfRelation=>

### 3.4.5.1. Spaces REST service

| Name                     | Service URL                                  | Location                                                                                                  | Description                                                                         |
|--------------------------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| <b>SpacesRestService</b> | {restContextName}/{portalName}/social/spaces | <b>Maven groupId:</b><br>org.exoplatform.social<br><br><b>ArtifactId:</b><br>exo.social.component.service | Provides REST services for space gadget to display user's spaces and pending spaces |

#### • API:

| Name                   | Service URL                                    | Parameters | Expected Values | Description |
|------------------------|------------------------------------------------|------------|-----------------|-------------|
| <b>showMySpaceList</b> | {restContextName}/social/spaces/mySpaces/show. |            | {format} String | Shows       |

| Name                        | Service URL<br>Endpoint | Parameters                                                                                                | Expected Values                                                             | Description                                                       |  |
|-----------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|-------------------------------------------------------------------|--|
|                             |                         | portalName<br><br>format                                                                                  | String<br><br>String: json or xml                                           | mySpaceList by the json/xml format.                               |  |
| <b>showPendingSpaceList</b> | {restContextName}       | /social/spaces/pendingSpaces/show.<br>portalName<br><br>format                                            | String<br><br>String: json or xml                                           | <b>Shows</b> {pendingSpaceList} by the json/xml format.           |  |
| <b>suggestSpacenames</b>    | {restContextName}       | /social/spaces/suggest.<br>portalName<br><br>conditionToSearch<br>typeOfRelation<br>currentUser<br>format | {format}<br>String<br><br>String<br>String<br>String<br>String: json or xml | Returns space's names that match the input string for suggesting. |  |

Example:

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

### 3.4.5.2. Widget REST service

| Name                     | Service URL               | Location                                                                                                                      | Description                                                                |
|--------------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <b>WidgetRestService</b> | {restContextName}/spaces/ | {containerName}<br><b>Maven groupId:</b><br>org.exoplatform.social<br><br><b>ArtifactId:</b><br>exo.social.extras.widget.rest | Provides REST services for creating spaces or getting spaces' information. |

- **API:**

| Name             | Service URL<br>Endpoint                  | Parameters                    | Expected Values | Description                                              |
|------------------|------------------------------------------|-------------------------------|-----------------|----------------------------------------------------------|
| <b>goToSpace</b> | {restContextName}/spaces/{containerName} | /go_to_space<br>containerName | String          | Creates (if not existing) or accesses a space. Two query |

| Name             | Service Endpoint                                    | URL           | Parameters                                     | Expected Values                                             | Description                                                                                                           |
|------------------|-----------------------------------------------------|---------------|------------------------------------------------|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
|                  |                                                     |               | portalName<br><br>spaceName<br><br>description | String (Default Value: classic)<br><br>String<br><br>String | parameters needed: spaceName and description                                                                          |
| <b>spaceInfo</b> | {restContextName}/spaces/{containerName}/space_info | containerName | portalName<br><br>spaceName<br><br>description | String<br><br>String (Default Value: classic)<br><br>String | Returns the HTML page for displaying the information of space. Two query parameters needed: spaceName and description |

Example:

[http://localhost:8080/rest-socialdemo/spaces/socialdemo/go\\_to\\_space?name=Social&description=Social](http://localhost:8080/rest-socialdemo/spaces/socialdemo/go_to_space?name=Social&description=Social)

[http://localhost:8080/rest-socialdemo/spaces/socialdemo/space\\_info?name=Social&description=Social](http://localhost:8080/rest-socialdemo/spaces/socialdemo/space_info?name=Social&description=Social)

## 3.5. Sample code/tutorial

### 3.5.1. Activity Stream

eXo Social provides a way to share status updates and activity information for users as well as spaces (aka Activity Streams). With the API, you can customize the activities or publish new ones.

To manipulate activities, you will use the [ActivityManager](#). To get an instance of this class, you will need to use the PortalContainer.

#### 3.5.1.1. Create an activity

There are two types of activities : activities for a user and activities for a space. The following examples will show you how to create an activity for each type.

#### 3.5.1.2. Publish an activity for a user

Users have activity streams. The code below shows you how to publish a nex activity into the public activity stream of a user.

```
import org.exoplatform.container.PortalContainer;
import org.exoplatform.social.core.activitystream.ActivityManager;
import org.exoplatform.social.core.activitystream.model.Activity;
import org.exoplatform.social.core.identity.IdentityManager;
```

```

import org.exoplatform.social.core.identity.impl.organization.OrganizationIdentityProvider;
import org.exoplatform.social.core.identity.model.Identity;

public void createActivityForUser() {
    String username = "zun";
    // Get current container
    PortalContainer container = PortalContainer.getInstance();

    // Get IdentityManager to handle identity operation
    IdentityManager identityManager = (IdentityManager) container.getComponentInstance(IdentityManager.class);

    // Get ActivityManager to handle activity operation
    ActivityManager activityManager = (ActivityManager) container.getComponentInstanceOfType(ActivityManager.class);

    // Get existing user or create a new one
    try {
        Identity userIdentity = identityManager.getOrCreateIdentity(OrganizationIdentityProvider.NAME, username);

        // Create new activity for this user
        Activity activity = new Activity();
        activity.setUserId(userIdentity.getId());
        activity.setTitle("Hello World!");
        // Save activity into JCR using ActivityManager
        activityManager.saveActivity(activity);
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

### 3.5.1.3. Publish an activity for a space

Spaces are also social objects and thus, they own an activity stream. The code below shows you how to publish activity into a space's stream

```

import org.exoplatform.container.PortalContainer;
import org.exoplatform.social.core.activitystream.ActivityManager;
import org.exoplatform.social.core.activitystream.model.Activity;
import org.exoplatform.social.core.identity.IdentityManager;
import org.exoplatform.social.core.identity.impl.organization.OrganizationIdentityProvider;
import org.exoplatform.social.core.identity.model.Identity;
import org.exoplatform.social.space.Space;
import org.exoplatform.social.space.SpaceException;
import org.exoplatform.social.space.SpaceService;
import org.exoplatform.social.space.impl.SpaceIdentityProvider;

//...

public void createActivityForSpace() {
    //make sure a space with name "mySpace" is created.
    String spaceName = "mySpace";
    String username = "zun";
    // Get current container
    PortalContainer container = PortalContainer.getInstance();

    // Get IdentityManager to handle identity operation
    IdentityManager identityManager = (IdentityManager) container.getComponentInstance(IdentityManager.class);

    // Get ActivityManager to handle activity operation
    ActivityManager activityManager = (ActivityManager) container.getComponentInstanceOfType(ActivityManager.class);

    // Get SpaceService to handle space operation
    SpaceService spaceService = (SpaceService) container.getComponentInstanceOfType(SpaceService.class);
    try {
        Space space = spaceService.getSpaceByName(spaceName);
        if (space != null) {
            // Get space identity via SpaceIdentityProvider
            Identity spaceIdentity = identityManager.getOrCreateIdentity(SpaceIdentityProvider.NAME, spaceName);
            // Get identity instance of the user who wants to create activity
            Identity userIdentity = identityManager.getOrCreateIdentity(OrganizationIdentityProvider.NAME, username);
            // Create new activity for this space
            Activity activity = new Activity();
            activity.setUserId(userIdentity.getId());

```

```

        activity.setTitle("An activity for space");
        activityManager.saveActivity(spaceIdentity, activity);
    }
} catch (SpaceException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
}

```

### 3.5.1.4. Useful functions

- [ActivityManager#getActivity](#)
- [ActivityManager#getActivities](#)
- [ActivityManager#saveActivity](#)
- [ActivityManager#saveComment](#)
- [ActivityManager#saveLike](#)
- [ActivityManager#removeLike](#)
- [ActivityManager#getComments](#)

#### 3.5.1.4.1. Configure an activity processor

An Activity processor is used for modifying the content of activities before they are responded and rendered at client's browser. For example, we will create an activity processor for replacing all the texts representing the smile face ":-)" in the activity title by the smiley icons.

Firstly, we will create the SmileyProcessor class by extending the BaseActivityProcessorPlugin

```

package org.exoplatform.social.core.activitystream;

public class SmileyProcessor extends BaseActivityProcessorPlugin {
    public SmileyProcessor(InitParams params) {
        super(params);
    }

    String smiley = "<img src={{\"\"}}http://www.tombraider4u.com/pictures/smiley.gif{{\"\"}}/>";

    public void processActivity(Activity activity) {
        String title = activity.getTitle();
        activity.setTitle(title.replaceAll(":-)", smiley));
    }
}

```

And then, we have to register this processor by adding some XML configuration into the project configuration file (configuration.xml)

```

<component>
  <key>org.exoplatform.social.core.activitystream.ActivityManager</key>
  <type>org.exoplatform.social.core.activitystream.ActivityManager</type>
  <component-plugins>
    <component-plugin>
      <name>SmileyProcessor</name>
      <set-method>addProcessorPlugin</set-method>
    </component-plugin>
  </component-plugins>
</component>

```

```

<type>org.exoplatform.social.core.activitystrea.SmileProcessor</type>
<init-params>
  <values-param>
    <name>priority</name>
    <value>1</value>
  </values-param>
</init-params>
</component-plugin>
</component-plugins>
</component>

```

"init-params" contains all the key-value data which a processor will use to initialize. At the above config, priority value indicates the order that this processor will be used. So with '1' value, this processor will be used before all remaining processors with lower priority.

### 3.5.1.5. Publish an rss feed with feedmesh

It's really easy to publish an rss feed to a space's activity stream. eXo Social already provides FeedmeshJobPlugin for publishing rss feeds. As you can see in project `exo.social.extras.feedmesh`, there are JiraFeedConsumer and HudsonFeedConsumer samples to post eXo Social project's feeds (jira and hudson) to a pre-defined space: exosocial in a specifiportal container: socialdemo as in the configuration file:

```

<external-component-plugins>
  <target-component>org.exoplatform.services.scheduler.JobSchedulerService</target-component>
  <component-plugin>
    <name>RepubSocialJiraActivityJob</name>
    <set-method>addPeriodJob</set-method>
    <type>org.exoplatform.social.feedmesh.FeedmeshJobPlugin</type>
    <description/>
    <init-params>
      <properties-param>
        <name>mesh.info</name>
        <property name="feedURL" value="http://jira.exoplatform.org/plugins/servlet/streams?key=SOC"/>
        <property name="categoryMatch" value="resolved|created"/>
        <property name="targetActivityStream" value="space:exosocial"/>
        <property name="portalContainer" value="socialdemo"/>
      </properties-param>
      <properties-param>
        <name>job.info</name>
        <description>save the monitor data periodically</description>
        <property name="jobName" value="JIRAFeedConsumer"/>
        <property name="groupName" value="Feedmesh"/>
        <property name="job" value="org.exoplatform.social.feedmesh.JiraFeedConsumer"/>
        <property name="repeatCount" value="0"/>
        <property name="period" value="60000"/>
        <property name="startTime" value="+45"/>
        <property name="endTime" value=""/>
      </properties-param>
    </init-params>
  </component-plugin>
  <component-plugin>
    <name>WatchSocialBuildStatus</name>
    <set-method>addPeriodJob</set-method>
    <type>org.exoplatform.social.feedmesh.FeedmeshJobPlugin</type>
    <description/>
    <init-params>
      <properties-param>
        <name>mesh.info</name>
        <property name="feedURL" value="http://builder.exoplatform.org/hudson/view/social/job/social-trunk-ci"/>
        <property name="targetActivityStream" value="space:exosocial"/>
        <property name="portalContainer" value="socialdemo"/>
      </properties-param>
      <properties-param>
        <name>job.info</name>
        <description>save the monitor data periodically</description>
        <property name="jobName" value="HudsonFeedConsumer"/>
        <property name="groupName" value="Feedmesh"/>
        <property name="job" value="org.exoplatform.social.feedmesh.HudsonFeedConsumer"/>
        <property name="repeatCount" value="0"/>
        <property name="period" value="60000"/>
        <property name="startTime" value="+100"/>
      </properties-param>
    </init-params>
  </component-plugin>

```

```

        <property name="endTime" value=""/>
    </properties-param>
</init-params>
</component-plugin>
</external-component-plugins>

```

When running eXo Social, login with <http://localhost:8080/socialdemo> and create a space named "exosocial". Done, all the feeds from jira and hudson for Social project will be automatically published to exosocial space.

## 3.5.2. People

eXo Social provides a way to add profile informations, relationships and connections between users: People. With the eXo People API, profile informations and relationship can be easily managed and customized.

### 3.5.2.1. Identity

The identity allows to identity uniquely a social object. Social objects can be persons, groups, applications, or whatever you think deserves social interactions like connecting, publishing an activity stream or holding a profile.

#### 3.5.2.1.1. IdentityProvider

Creating your Identity Provider allows you to integrate people outside of your portal (for exemple customers) into your social network without having to create a portal account. You can also use this to populate the profile with data coming from other systems. Here is an example :

```

class SampleIdentityProvider extends OrganizationIdentityProvider{
    public SampleIdentityProvider(OrganizationService organizationService) {
        super(organizationService);
    }

    @Override
    public void populateProfile(Profile profile, User user) {
        profile.setProperty(Profile.FIRST_NAME, "this is first name");
        profile.setProperty(Profile.LAST_NAME, "this is last name");
        profile.setProperty(Profile.USERNAME, "this is user name");
        profile.setProperty(Profile.URL, "/path/to/profile/");
    }
}

```

In this example, we created a SampleIdentityProvider class extending OrganizationIdentityProvider which is the IdentityProvider used to connect to the portal user's base. In this class, we overrided the populateProfile method and added some dummy data in the profile fields.

#### 3.5.2.1.2. IdentityManager

The IdentityManager is the service used to manipulate the identity operations like creating, getting, deleting or finding a profile. We can get the IdentityManager via the ExoContainer. The following code will show how to get an IdentityManager instance and create a basic identity instance :

```

import org.exoplatform.container.ExoContainer;
import org.exoplatform.container.ExoContainerContext;
import org.exoplatform.social.core.identity.IdentityManager;
import org.exoplatform.social.core.identity.impl.organization.OrganizationIdentityProvider;
import org.exoplatform.social.core.identity.model.Identity;

//.....

String containerName = "portal";
String username = "zun";

```

```
//get container to get other registered components
ExoContainer container = ExoContainerContext.getContainerByName(containerName);

//get IdentityManager to handle identity operation
IdentityManager identityManager = (IdentityManager) container.getComponentInstanceOfType(IdentityManager.class);

//get ActivityManager to handle activity operation
ActivityManager activityManager = (ActivityManager) container.getComponentInstanceOfType(ActivityManager.class);

//create new user with name Zun
Identity userIdentity = identityManager.getOrCreateIdentity(OrganizationIdentityProvider.NAME, username);
```

### 3.5.2.2. ProfileListener

The People API provides some notification interfaces which programmers can implement in order to create their own handlers for notifications like notifications() for profile modifications([ProfileListenerPlugin](#)) or for relationship changes([RelationshipListenerPlugin](#)). The following example will guide you through implementing one of these interfaces and show you how to configure this plugin.

We will create the class ProfileLoggerListener. Its tasks is to log all profile modifications of the systems. The abstract class ProfileListenerPlugin provides us the interface to implement this method.

```
import org.exoplatform.services.log.ExoLogger;
import org.exoplatform.services.log.Log;
import org.exoplatform.social.core.identity.lifecycle.ProfileListenerPlugin;
import org.exoplatform.social.core.identity.spi.ProfileLifeCycleEvent;

public class ProfileLoggerListener extends ProfileListenerPlugin{
    private static final Log logger = ExoLogger.getExoLogger(ProfileLoggerListener.class);
    @Override
    public void avatarUpdated(ProfileLifeCycleEvent event) {
        logger.info("@ " + event.getUsername() + " profile has updated his basic profile info.");
    }

    @Override
    public void basicInfoUpdated(ProfileLifeCycleEvent event) {
        logger.info("@ " + event.getUsername() + " profile has updated his basic profile info.");
    }

    @Override
    public void contactSectionUpdated(ProfileLifeCycleEvent event) {
        logger.info("@ " + event.getUsername() + " profile has updated his contact info.");
    }

    @Override
    public void experienceSectionUpdated(ProfileLifeCycleEvent event) {
        logger.info("@ " + event.getUsername() + " profile has an updated experience section.");
    }

    @Override
    public void headerSectionUpdated(ProfileLifeCycleEvent event) {
        logger.info("@ " + event.getUsername() + " has updated his header info.");
    }
}
```

After creating the ProfileLoggerListener class, we have to add some configurations for this class to the configuration.xml :

```
<external-component-plugins>
  <target-component>org.exoplatform.social.core.identity.IdentityManager</target-component>
  <component-plugin>
    <name>ProfileLoggerListener</name>
    <set-method>addProfileListener</set-method>
    <type>path.to.ProfileLoggerListener</type>
  </component-plugin>
</external-component-plugins>
```

### 3.5.2.3. Relationships

Similarly, you can apply the above steps to implement the `RelationshipListenerPlugin` for relationship notifications.

Relationship is the bridge between two identities in eXo Social. There are many types of relationships defined in the `Relationship` class. With these types, a user can invite another user, confirm invitations or remove relationship.

#### 3.5.2.3.1. User connection

The following code will show how to invite a user to connect to another :

```
import org.exoplatform.container.ExoContainer;
import org.exoplatform.container.ExoContainerContext;
import org.exoplatform.social.core.identity.IdentityManager;
import org.exoplatform.social.core.identity.impl.organization.OrganizationIdentityProvider;
import org.exoplatform.social.core.identity.model.Identity;
import org.exoplatform.social.core.relationship.Relationship;
import org.exoplatform.social.core.relationship.RelationshipManager;

public void inviteUser() throws Exception {
    String containerName = "portal";
    ExoContainer container = ExoContainerContext.getContainerByName(containerName);

    IdentityManager identityManager = (IdentityManager) container.getComponentInstanceOfType(IdentityManager.class);

    Identity invitedIdentity = identityManager.getOrCreateIdentity(OrganizationIdentityProvider.NAME, "Hoat");
    String invitedUserId = invitedIdentity.getId();

    String currUserId = "Zun";
    Identity currentIdentity = identityManager.getOrCreateIdentity(OrganizationIdentityProvider.NAME, currUserId);

    RelationshipManager relationshipManager = (RelationshipManager) container.getComponentInstanceOfType(RelationshipManager.class);
    Relationship relationship = relationshipManager.invite(currentIdentity, invitedIdentity);
}
```

#### 3.5.2.3.2. RelationshipListener

Let's take a look at relationship Listener([RelationshipListenerPlugin](#)). The following example will show you how to implement one of these interfaces and how to configure this plugin. The following step is similar to the Profile notifications implementation.

```
import org.exoplatform.social.core.relationship.Relationship;
import org.exoplatform.social.core.relationship.lifecycle.RelationshipListenerPlugin;
import org.exoplatform.social.relationship.spi.RelationshipEvent;

public class RelationshipLoggerListener extends RelationshipListenerPlugin {
    private static final Log logger = ExoLogger.getExoLogger(RelationshipLoggerListener.class);

    @Override
    public void confirmed(RelationshipEvent event) {
        String[] names = getUserNamesFromEvent(event);
        logger.info(names[0] + " confirmed the invitation of " + names[1]);
    }

    @Override
    public void ignored(RelationshipEvent event) {
        String[] names = getUserNamesFromEvent(event);
        logger.info(names[0] + " ignored the invitation of " + names[1]);
    }

    @Override
    public void removed(RelationshipEvent event) {
        String[] names = getUserNamesFromEvent(event);
    }
}
```

```

    logger.info(names[0] + " removed the relationship with " + names[1]);
}

private String[] getUserNamesFromEvent(RelationshipEvent event){
    Relationship relationship = event.getPayload();

    Identity id1 = relationship.getIdentity1();
    Identity id2 = relationship.getIdentity2();

    String user1 = "@" + id1.getRemoteId();
    String user2 = "@" + id2.getRemoteId();

    return new String[]{user1, user2 };
}
}

```

After creating the RelationshipLoggerListener class, we have to add some configurations for this class to the configuration.xml :

```

<external-component-plugins>
  <target-component>org.exoplatform.social.core.relationship.RelationshipManager</target-component>
  <component-plugin>
    <name>RelationshipLoggerListener</name>
    <set-method>addListenerPlugin</set-method>
    <type>classpath.of.your.RelationshipLoggerListener</type>
  </component-plugin>
</external-component-plugins>

```

### 3.5.3. Spaces

eXo Social provides a way to create groups and to share data and applications: the space. A space has it's own activity stream in which applications or members can publish information. In each space, members share applications and an openSocial dashboard.

To manipulate the spaces, you will use the SpaceService. To get an instance of this class, you will need to get current PortalContainer instance.

#### 3.5.3.1. Spaces Management

##### 3.5.3.1.1. Create a space

The following example shows how to create a space:

```

package org.exoplatform.social.sample;

import org.exoplatform.container.PortalContainer;
import org.exoplatform.social.application.impl.DefaultSpaceApplicationHandler;
import org.exoplatform.social.space.Space;
import org.exoplatform.social.space.SpaceException;
import org.exoplatform.social.space.SpaceService;

public class SpaceCreationSample {

    public void createSpace() throws SpaceException {
        String spaceName = "mySpace";
        String creator = "jeremi";
        PortalContainer container = PortalContainer.getInstance();
        SpaceService spaceService = (SpaceService) container.getComponentInstanceOfType(SpaceService.class);
        // We verify if there is no space already created
        Space space = spaceService.getSpaceByName(spaceName);
        if (space == null) {
            space = new Space();
            space.setName(spaceName);
            space.setRegistration(Space.OPEN);
            space.setDescription("space description");
            //DefaultSpaceApplicationHandler is the default implementation of SpaceApplicationHandler. You can create y

```

```

        space.setType(DefaultSpaceApplicationHandler.NAME);
        //We create the space
        space = spaceService.createSpace(space, creator);
        //We initialize the applications
        spaceService.initApp(space);
    }
}
}

```

### 3.5.3.2. Space's applications management

#### 3.5.3.2.1. Add an application to a space

We can add a portlet or gadget applications into spaces. Once added, all members of the space can use that application. The following code shows how to add an application into a space :

```

public void addApplicationToSpace() throws SpaceException {
    //Your portlet name
    String appId = "AnyGadgetOrAnyPortletName";
    String spaceId = "zunSpace";

    //get container to get other registered components
    PortalContainer container = PortalContainer.getInstance();

    //get space service for installing operations
    SpaceService spaceService = (SpaceService) container.getComponentInstanceOfType(SpaceService.class);

    //install application for the space
    spaceService.installApplication(spaceId, appId);

    //we must activate installed application to be able to use it
    spaceService.activateApplication(spaceId, appId);
}

```



#### Note

appId is the portlet or gadget name as defined in portlet.xml or gadget.xml in the web-app. With eXo Social, we can find it in social-portlet.war and social.war.

### 3.5.3.3. Members Management

The SpaceService allows you to manage the spaces' members. Here is how you would add a new member to a space:

```

String spaceName = "mySpace";

PortalContainer container = PortalContainer.getInstance();
SpaceService spaceService = (SpaceService) container.getComponentInstanceOfType(SpaceService.class);

Space space = service.getSpaceByName(spaceName);
if (space != null) {
    spaceService.addMember(space, "mary");
}

```

### 3.5.3.4. Listener to a space lifecycle

To receive notifications of what is happening in spaces, you need to extend SpaceListenerPlugin and register it. Every method takes a SpaceLifeCycleEvent object as parameter that contain the information about the event and its context.

The events available are:

- `spaceCreated`: this event is called right after space is created successfully with its applications.
- `spaceRemoved`: this event is called right after a space is removed: application's removed, its group and group navigation is removed
- `applicationAdded`: this event is called right after an application is added (installed) to a space
- `applicationActivated`: this event is called right after an application is activated.
- `applicationDeactivated`: this event is called right after an application is deactivated.
- `applicationRemoved`: this event is called right after an application is removed from a space.
- `joined`: this event is called right after a user joins a space.
- `left`: this event is called right after a space member leaves its space.
- `grantedLead`: this event is called right after a user is granted space's manager role.
- `revokedLead`: this event is called right after a space's manager is revoked his role to be a space member.

```
import org.exoplatform.social.space.lifecycle.SpaceListenerPlugin;

public class MySpaceListenerPlugin extends SpaceListenerPlugin {
    ...
}
```

As an example, see the [SpaceActivityPublisher](#) that publishes an activity based on an event that happened in a space.

To register your listener, hook it to the `SpaceService` like this :

```
<external-component-plugins>
  <target-component>org.exoplatform.social.space.SpaceService</target-component>
  <component-plugin>
    <name>SpaceActivityPublisher</name>
    <set-method>addSpaceListener</set-method>
    <type>org.mycompany.MySpaceListenerPlugin</type>
  </component-plugin>
</external-component-plugins>
```

## 3.5.4. Open Social

eXo Social is implementing the OpenSocial standard. So you can integrate OpenSocial Gadget in your dashboard and use the RPC or REST API to access to the social data. With the support for the OpenSocial standard, eXo Social provides a framework for developers to build gadgets that can display and mash-up activity information for contacts, social networks, applications and services.

### 3.5.4.1. Gadget

Gadgets are web-based software components based on HTML, CSS, and JavaScript. They allow developers to easily write useful web applications that work anywhere on the web without modification. To know more, we suggest some links for detail information :

## [Gadgets Specification](#)

### [OpenSocial Core Gadget Specification 10](#)

After getting acquainted with Gadget concept, we enter into the detail on how to create an opensocial gadget. However, the tutorials are done greatly in Opensocial official site so we refer you to read these tutorials at [that website](#)

We only note that in Opensocial gadgets only work in the dashboard in eXo Social.

#### **3.5.4.1.1. Supported APIs**

As we have said above, eXo Social is implementing the OpenSocial standard. So every eXo Social implementations apply the Opensocial Specification generally or [Apache Shindig](#) specifically. Therefore, eXo Social uses and extends Apache Shindig APIs to compatible with the common Opensocial APIs which is supported by other big social networks like [Ning](#), [Hi5](#), [Orkut](#) ...

To get more detail about Supported APIs, we refer you to read [Opensocial Specs](#)

#### **3.5.4.2. REST/RPC API**

If your eXo social server is running on <http://localhost:8080/> the address of the API will be:

REST API: <http://localhost:8080/social/social/rest>

RPC API: <http://localhost:8080/social/social/rpc>

To learn what you can do with this APIs, have a look at the [specification](#). If you are developing in Java, you can use the [opensocial-java-client](#)

##### **3.5.4.2.1. Configure the security**

If you are using opensocial, there is good chance you are going to need to configure the oAuth authentication. To do this, you need to edit the configuration to add you oAuth key.

Edit the file: `gatein/conf/portal/portal/configuration.xml` and add this component:

```
<component>
  <key>org.exoplatform.social.opensocial.oauth.ServiceProviderStore</key>
  <type>org.exoplatform.social.opensocial.oauth.ServiceProviderStore</type>

  <init-params>
    <properties-param>
      <name>grails-book-flow</name>
      <description>consmer key and secret for sample oauth provider. </description>
      <property name="consumerKey" value="YOUR_KEY_HERE" />

      <property name="sharedSecret" value="YOUR_SECRET_KEY_HERE" />
    </properties-param>
  </init-params>
</component>
```

The consumerKey and sharedSecret are the key that need to be shared with the application that is doing the request.

##### **3.5.4.2.2. Publish an activity into a space**

eXo Social added this functionality that is not available in the standard opensocial API. You can publish activities into a space using the opensocial API.

To do this, instead of publishing your activity to the group @self as usual, publish it to the group "space:spaceID" or "space:spaceName".

Using the opensocial java library and groovy, your code will look like this:

```
def client = getOpenSocialClient()

//we create our new activity
Activity activity = new Activity()
activity.title = "BookFlow Purchase"
activity.body = "xx purchased the book xxx"

//We prepare the request that will create the activity
Request request = ActivitiesService.createActivity(activity);
//We specify that the creation of this new activity is for the space bookflow
request.groupId = "space:bookflow";

client.send(request);
```

As you can see in this example, we set the groupId to "space:bookflow", bookflow being the name of our space.

#### 3.5.4.2.2.1. Tutorial

- See [Grails + eXo Social tutorial](#).

### 3.5.5. Space widget tutorial

The eXo Social widget enables developers to add capabilities of eXo Social to external applications. Since the widget is hosted on your eXo Social server, it can display personalized information. An activity stream of the most recent user actions will display to the group's members.

There are two options of the eXo Social widget that provide different levels of integration and information.

The basic version of this widget is an iFrame. The more advanced version is a button you can insert in a page; this will display a small pop-up with information about the space.

#### 3.5.5.1. Basic version

To insert the basic version, you need to have an iFrame to insert on your site.

```
<iframe scrolling="no" height="180" frameborder="no" width="220" src="http://URL_OF_YOUR_EXO_INSTALLATION.COM/r
```

To install this version in your application, replace all the uppercase text below:

- URLOFYOUREXOINSTALLATION.COM - this is the URL of your eXo Social installation. If you are testing on your local computer, the URL may be *localhost:8080\_*
- NAMEOFOURSPACE - *this is the title of your space in eXo Social. In the URL, it is necessary to avoid special characters. For the space name, you can only use alphanumeric characters and "'", ".", "-" or " "*
- DESCRIPTIONOFTHESPACE - *this will be displayed in the list of spaces.*

### 3.5.5.2. Advanced version

To install an advanced version of the widget, you need to insert a code snippet in your page. This includes some HTMLs plus some JavaScripts. The necessary CSS will be added dynamically.

Next, insert the following code at the position you want the button to be displayed:

```
<div class="exoSpacesContainer"><a href="javascript:void(0);" id="exoSpacesLink" class="exoSpacesLink" target="_blank">Space</a><script src="/socialWidgetResources/javascript/space.js"></script><script>spaces.createPopup("exoSpacesLink", "MyAppName - my social object", "my cool description");</script></div>
```

The important function here is:

```
spaces.createPopup(link, spaceName, description)
```

- **link:** link is the ID or the HTML element where the pop-up will be placed. If you copy and paste the code snippet provided above, you don't need to change this value.
- **spaceName:** This is the name of space. It is also used to identify the space. We recommend you use the following format: "MyAppName - my social object"
- **description:** This is the description of your space that will be displayed within eXo Spaces. It is used when a new space is created.

### 3.5.5.3. Configure

- **serverURL** (Default: "<http://127.0.0.1:8080>"): The address of your eXo installation. To change it, use `spaces.setServerURL(...)`;
- **spaceServicePath** (Default: `"/rest/private/spaces/"`): The path to the spaces service. It is rare you have to change it; but if needed, use `spaces.setSpaceServicePath(...)`;
- **portalName** (Default: `"socialdemo"`): The name of portal you are using. To change it, use `spaces.setPortalName(...)`;

If you want to change any part of this configuration, the best way is to change before creating the pop-up. For example:

```
<div class="exoSpacesContainer"><a href="#" id="exoSpacesLink" class="exoSpacesLink" target="_blank">Space</a><script src="/socialWidgetResources/javascript/space.js"></script><script>spaces.setServerURL("http://192.168.2.100:8080");spaces.createPopup("exoSpacesLink", "My cool new space", "my cool description");</script></div>
```

You can see an example of integration at: <http://localhost:8080/socialWidgetResources/test.html>

## 3.6. Public Javascript APIs

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