eXo Platform 3.0 - Collaboration Functions

Chapter 1. Prerequisites

You can refer to the following links to understand more about the eXo Collaboration Reference Guide:

- GateIn Reference Guide
- OpenSocial gadget
- WebService
- Chatserver
- Vcard
- iCalendar and its Internet standard

Chapter 2. Applications

2.1. Portlets

2.1.1. Calendar portlet

The Calendar portlet is packaged in the *calendar.war* file.

2.1.1.1. Description

The Calendar portlet shows the Calendar application of eXo Collaboration with a lot of features provided to users.

The Calendar application includes the following features:

- Create multiple personal calendars, manage calendars easily with calendar groups.
- Create events or tasks using the **Quick Add** dialog easily.
- Create events or tasks in details.
- Create all-day events.
- View other attender's availability schedules.
- Create recurring events.
- Get reminders.
- View calendars by various views: day, week, month and year.
- View events day-by-day by navigating the mini-calendar quickly.
- Share calendars with others.
- Import/Export calendars.
- Publish your calendars with RSS, CalDAV.
- Search for events and/or tasks in calendars.
- Print your agenda.

2.1.1.2. Portlet.xml

To see the portlet in the project, follow this path: */eXoApplication/calendar/webapp/src/main/webapp/WEB-INF/portlet.xml*.

2.1.2. Chatbar portlet

The Chatbar portlet is packaged in the *Chatbar.war* file.

2.1.2.1. Description

The Chatbar portlet shows the Chatbar application of eXo Collaboration that can be positioned in the portal or page layout as any other, but behaves as a floating box. The bar remains floating at its location even when the browser window is scrolled or resized. Its height is fixed, but can be expanded horizontally to any size available in its container. This allows the portlet to be placed in two layout cases:

- Large width area (typically header or footer).
- Narrow column.

The Chatbar application implements all functions of the Chat application, allowing you to send and receive messages anywhere after you are logged in successfully. The Chatbar is a typical toolbar with buttons to open menus. It gives access to main features of Chat:

- Status change and presence indicator.
- Contacts.
- Rooms.
- Minimized conversation window.

2.1.2.2. Portlet preferences

The Chatbar Portlet consists of some preferences as in the following sample code:

```
<portlet-preferences>
  <preference>
    <name>showMailLink</name>
    <value>true</value> <!--true/false -->
    <read-only>false</read-only>
  </preference>
  <preference>
    <name>showCalendarLink</name>
    <value>true</value> <!--true/false -->
    <read-only>false</read-only>
  </preference>
  <preference>
    <name>showContactLink</name>
    <value>true</value> <!--true/false -->
    <read-only>false</read-only>
  </preference>
  <preference>
    <name>mailUrl</name>
    <value>portal/private/intranet/mail</value> <!--String page name -->
    <read-only>false</read-only>
  </preference>
  <preference>
    <name>calendarUrl</name>
    <value>portal/private/intranet/calendar</value> <!--String page name -->
    <read-only>false</read-only>
  </preference>
  <preference>
    <name>contactUrl</name>
    <value>portal/private/intranet/contact</value> <!--String page name -->
    <read-only>false</read-only>
  </preference>
  <preference>
    <name>info</name>
```

```
    <value>info</value> <!--this is only the key to get the resource bundle the full key : UIConfigForm.label.in
    <read-only>true</read-only>
  </preference>
</portlet-preferences>
```

In which:

Preference Name	Possible Values	Default Values	Description
showMailLink	`true / false`	`true`	The value as "true" or "false" means that users are allowed to see the application icon or not respectively.
showCalendarLink	`true / false`	`true`	The value as "true" or "false" means that users are allowed to see the application icon or not respectively.
showContactLink	`true / false`	`true`	The value as "true" or "false" means that users are allowed to see the application icon or not respectively.
mailUrl	`string`	`Portal/private/intranet/mail`	The URL to the Mail application page in the portal without combining with the %domain name. The port% chatbar will resolve it from server.
calendarUrl	`string`	`Portal/private/intranet/calendar`	The URL to the Calendar application page in the portal without combining with the %domain name. The port% chatbar will resolve it from server.
contactUrl	`string`	`Portal/private/intranet/contact`	The URL to the Address Book application page in the portal without combining with the %domain name. The port% chatbar will resolve it from server.
info	`Info`	`Info`	This is only the key to get the resource bundle of the full key: UIConfigForm.label.info.

2.1.2.3. Portlet.xml

See the portlet in the project following this path: */eXoApplication/chatbar/webapp/src/main/webapp/WEB-INF/portlet.xml*.

2.1.3. Chat Portlet

The Chat portlet is packaged in the *Chat.war* file.

2.1.3.1. Description

The Chat Portlet shows the Chat application of eXo Collaboration that allows users to enter chat rooms and communicate with online others at real time.

2.1.3.2. Portlet.xml

See the portlet in the project following this path: */eXoApplication/chat/webapp/src/main/webapp/WEB-INF/portlet.xml*

2.1.4. Contact Portlet

Contact Portlet is packaged in the *Contact.war* file.

2.1.4.1. Description

Contact Portlet shows the Contact application of eXo Collaboration that allows users to personalize their contact view from different view types, such as List view and VCards view.

2.1.4.2. Portlet.xml

See the portlet in the project following this path: */eXoApplication/contact/webapp/src/main/webapp/WEB-INF/portlet.xml*.

2.1.5. Mail Portlet

The Mail Portlet is packaged in the *Mail.war* file.

2.1.5.1. Description

Mail Portlet shows the Mail application of eXo Collaboration that offers a lot of features to users such as sending, receiving or viewing their mails through Internet without actually downloading them to their computer. Users not only take advantages of eXo Mail by keeping and receiving all important messages, files and pictures forever but also by looking for and viewing their needed messages easily whenever they want. Additionally, the Mail application is smoothly integrated with other Collaboration modules, such as Address Book and Calendar.

2.1.5.2. Portlet.xml

See the portlet in the project following this path: */eXoApplication/mail/webapp/src/main/webapp/WEB-INF/portlet.xml*.

2.1.6. RSSreader Portlet

The RSSreader Portlet is packaged in the *Rssreader.war* file.

2.1.6.1. Description

eXo Collaboration uses the RSS Reader Portlet that facilitates users to quickly get a view of their favorite feeds around the web. They will get the latest news, the last updated posts from their favorite blogs, latest emails, and more.

2.1.6.2. Portlet.xml

See the portlet in the project following this path: */eXoApplication/content/webapp/src/main/webapp/WEB-INF/portlet.xml*.

2.2. Gadgets

eXo Collaboration consists of three gadgets: eventslist, taskslist and messageslist. They are packaged in the *csResources.war* file.

2.2.1. Eventslist

2.2.1.1. Description

Eventslist lists the maximum number of upcoming events, that is configurable by users. For example, they can set the preference list to 5 or 10 events.

See preferences of this gadget in the following sample code:

```
<UserPref datatype="string" display_name="__MSG_baseurl__" name="url" required="true" value="/calendar"/>
<UserPref datatype="string" display_name="__MSG_subscribeurl__" name="subscribeurl" required="true" value="/port
<UserPref datatype="string" default_value="10" display_name="__MSG_limit__" name="limit"/>
<UserPref datatype="enum" default_value="AM/PM" display_name="__MSG_format__" name="timeformat"/>
```

Details:

Preferences	Description
url	Link to the Calendar portlet.
Subscribeurl	Link to the upcoming events.
limit	The maximum number of upcoming events.
timeformat	The time format for upcoming events.

For more details on the preferences of gadgets, see here.

2.2.1.2. Links to used REST services

It uses the *upcomingEvent* service in the following package:

org.exoplatform.webservice.cs.calendar.CalendarWebservice.java.

2.2.2. Tasklist

Tasklist lists the maximum number of upcoming tasks that is configurable by users. For example, they can set the preference list to 5 or 10 tasks.

2.2.2.1. Description

See the preferences of this gadget in the following sample code:

```
<UserPref datatype="hidden" default_value="/calendar:/portal/rest/private/cs/calendar/upcoming:10:AM/PM:Default"
```

Accordingly, *setting* collects all the configuration of upcoming tasks and add some more functions to help developers change the configuration of the default skin.

2.2.2.2. Links to used REST services

It uses upcomingEvent service in the following package: *org.exoplatform.webservice.cs.calendar.CalendarWebservice.java*.

2.2.3. Messageslist

It lists the maximum number of unread messages, that is configurable by users.

2.2.3.1. Description

See the prefereces of this gadget in the following sample code:

```
<UserPref datatype="hidden" display_name="__MSG_baseurl__" name="url" required="true" value="/mail"/>
<UserPref datatype="hidden" display_name="__MSG_subscribeurl__" name="subscribeurl" required="true" value="/port
<UserPref datatype="hidden" default_value="5" display_name="__MSG_limit__" name="limit"/>
<UserPref datatype="hidden" default_value="" display_name="__MSG_account__" name="account"/>
<UserPref datatype="hidden" default_value="" display_name="__MSG_folder__" name="folder"/>
<UserPref datatype="hidden" default_value="" display_name="__MSG_tag__" name="tag"/>
```

Details:

Preferences	**Description**
Url	The URL of the Mail Application.
Subscribeurl	The link to upcoming messages.
Limit	The number of displayed unread messages that is set by users.
Account	The mail account in the Mail application.
Folder	The folder which consists of unread messages.
Tag	The tags in all unread messages.

2.2.3.2. Links to used REST services

It uses the unreadMail service in the following package: *org.exoplatform.webservice.cs.mail.MailWebservice.java*.

Chapter 3. Configurations

3.1. Components in eXo Collaboration Configuration

This table shows some main components that take init-param in the applications of eXo Collaboration:

Applications	Components	Description
Calendar	`CalendarServiceImpl`	It is a service that manages calendars in the Calendar application of eXo Collaboration.
Chat	`HistoryImpl`	It is a service that saves the chat history of users.
	`XMPPMessenger`	It is a service that processes messages of chat users, basing on the XMPP Protocol.
	`DefaultPresenceStatus`	It is a component that controls the presence status of chat users.
Contact	`ContactServiceImpl`	It is a service that supplies functions to manage contacts in the Address Book application of eXo Collaboration.
Webservice	`AddActionsPlugin`	It is used to register a listener for the actions on the nodes.

3.1.1. CalendarService

The configuration of the Calendar application is applied mainly in */eXoApplication/calendar/service/src/main/resources/conf/portal/configuration.xml*.

Use the CalendarService to configure the Calendar. The following information will explain details of its configuration. When this configuration file is executed, the component named *org.exoplatform.calendar.service.impl.CalendarServiceImpl* will process actions of the Calendar application.

```
<component>
        <key>org.exoplatform.calendar.service.CalendarService</key>
        <type>org.exoplatform.calendar.service.impl.CalendarServiceImpl</type>
        <init-params>
                <properties-param>
                        <name>eventNumber.info</name>
                        <property name="eventNumber" value="100"/>
                </properties-param>
        </init-params>
</component>
```

Details:

Properties-Param	Property name	Description	Possible Value	Default Value
eventNumber	`eventNumber`	The number of events in a calendar.	`interger`	`100`

3.1.2. HistoryImpl

The configuration of historyImpl is found in the */extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/chat/chat-service-configuration.xml*. When this configuration file is executed, the component named *org.exoplatform.services.xmpp.history.impl.jcr.HistoryImpl* initializes all the configured parameters.

```
<component>
        <type>org.exoplatform.services.xmpp.history.impl.jcr.HistoryImpl</type>
        <init-params>
                <value-param>
                        <name>workspace</name>
                        <value>collaboration</value>
                </value-param>
                <value-param>
                        <name>repository</name>
                        <value>repository</value>
                </value-param>
                <value-param>
                        <name>path</name>
                        <value>exo:applications/eXoChat/history</value>
                </value-param>
        </init-params>
</component>
```

Details:

Value-Param	Description	Possible Values	Default Value
workspace	The workspace name in JCR where history data is stored.	`string`	`collaboration`
repository	The repository name in JCR where history data is stored.	`string`	`repository`
path	The JCR path to the location where history data is stored.	`string`	`exo:applications/eXoChat/history`

3.1.3. XMPPMessenger

The configuration of the XMPPMessenger component is found in */extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/chat/chat-service-configuration.xml*. It helps eXo Collaboration connect the Openfire instance.

```
<component>
```

```
  <type>org.exoplatform.services.xmpp.connection.impl.XMPPMessenger</type>
  <init-params>
    <properties-param>
      <name>openfire-connection-conf</name>
      <property name="host" value="127.0.0.1"/>
      <property name="port" value="5222"/>
    </properties-param>
    <properties-param>
      <name>send-file</name>
      <property name="timeout" value="7200000"/>
    </properties-param>
  </init-params>
</component>
```

Details:

Properties-param	Property name	Description	Possible Values	Default Value

openfire-connection-conf	`host`
	
	`port`
		IP address or hostname for the openfire server.
		
		Port to connect on the Openfire server. Should be the same that is set in the Openfire configuration "Client to Server".
			`integer`
			
			`integer`
				127.0.0.1
				
				5222

send-file	`timeout`	The timeout before aborting attempt to establish a file transfer.	`integer`	`7200000`

3.1.4. DefaultPresenceStatus

The configuration of the DefaultPresenceStatus component is found in */extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/chat/chat-service-configuration.xml*.

```
<component>
  <type>org.exoplatform.services.presence.DefaultPresenceStatus</type>
  <init-params>
    <properties-param>
      <name>presence-status</name>
      <property name="mode" value="Free to chat"/>
    </properties-param>
  </init-params>
</component>
```

Details:

Properties-param	Property name	Description	Possible Values	Default Value
presence-status	`mode`	Show the present status of users.	`string`	`Free to chat`

3.1.5. ContactService

The configuration of the ContactService component is found in *eXoApplication/contact/service/src/main/resources/conf/portal/configuration.xml*.

When the server starts, the configuration file which contains the declaration of ContactService component is executed. A ContactService component is then created with params and plugins in the configuration file.

```
<component>
  <key>org.exoplatform.contact.service.ContactService</key>
  <type>org.exoplatform.contact.service.impl.ContactServiceImpl</type>
  <init-params>
    <values-param>
      <name>UserCanSeeAllGroupAddressBooks</name>
      <description>User can see all GroupAddressBooks or only GroupAddressBooks that the user has at least one m
      <value>false</value>
    </values-param>
    <values-param>
      <name>NonPublicGroups</name>
      <description>Groups that should not be displayed in broadcast list. Wildcards may be used in groups name</
    </values-param>
  </init-params>
</component>
```

Values-param	Description	Possible Values	Default Value
UserCanSeeAllGroupAddressBooks	User can see all GroupAddressBooks or only GroupAddressBooks that the user has at least one membership.	`true/false`	`false`
NonPublicGroups	Groups that should not be displayed in the broadcast list. Wildcards may be used in groups name.	`true/false`	N/A

3.2. External Component Plugins

The following table describes the main functions of external component plugins:

Applications	Components	Description
Calendar	`NewUserListener`	Create default personal calendars.
	`NewGroupListener`	Create default group calendars.
	`NewMembershipListener`	Share calendars to members of a specific group.
	`ReminderPeriodJob`	Execute sending reminder emails to users.
	`PopupReminderPeriodJob`	Open a pop-up reminder on the browser of users.
	`AddActionsPlugin`	Enable the systems to

Applications	Components	Description
		automatically update the updated date of events/tasks in a calendar when contents of these events/tasks are changed.
Chat	`HistoryPeriodJob`	Save the chat history of users.
	`RequestFilterComponentPlugin`	Delete the session of a user when the browser is suddenly closed or the session is changed.
	`AuthenticationLoginListener`	Start the session and log in the chat server.
	`AuthenticationLogoutListener`	End the session and log out the chat server.
Contact	`NewUserListener`	Create personal contact data for users.
	`NewMembershipListener`	Create address book for a specific group.
	`UpdateUserProfileListener`	Update the personal profile of a user when it is changed on the portal.
Content	`RSSContentPlugin`	The formatter used to analyze the data from a RSS resource.
	`DescriptionPlugin`	Represent the data from a RSS resource.
Mail	`AuthenticationLogoutListener`	Stop checking mails of a user when he logs out.
Social Intergration	`CalendarDataInitialize`	Create a calendar for a group in a specific space.
	`ContactDataInitialize`	Create an address book for a group in a specific space.
	`ContactSpaceActivityPublisher`	Customize the activity status of a specific space when an event happens on an address book.
	`CalendarSpaceActivityPublisher`	Customize the activity status of a specific space when an event happens on a calendar.
	`PortletPreferenceRequiredPlugin`	Declare the application that will automatically create database.

3.2.1. Calendar Configuration

3.2.1.1. NewUserListener

Each user can have a default personal calendar created. Use the NewUserListener to configure that. To use the plugin in the component configuration, you must use the target-component:

```
<target-component>org.exoplatform.services.organization.OrganizationService</target-component>
```

The configuration is applied mainly in *extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/calendar/calendar-service-configuration.xml*.

```
<component-plugin>
  <name>calendar.new.user.event.listener</name>
  <set-method>addListenerPlugin</set-method>
  <type>org.exoplatform.calendar.service.impl.NewUserListener</type>
  <description>description</description>
  <init-params>
    <value-param>
      <name>defaultEventCategories</name>
      <value>Meeting,Calls,Clients,Holiday,Anniversary</value>
    </value-param>
    <value-param>
      <name>defaultCalendarCategory</name>
      <value>My group</value><!-- Single value -->
    </value-param>
    <value-param>
      <name>defaultCalendar</name>
      <value>Default</value>
    </value-param>
    <!--Params for default calendar setting -->
    <value-param>
      <name>viewType</name>
      <value>1</value>
    </value-param>

    <value-param>
      <name>timeInterval</name>
      <value>15</value><!-- in minutes -->
    </value-param>

    <value-param>
      <name>weekStartOn</name>
      <value>2</value>
    </value-param>

    <value-param>
      <name>dateFormat</name>
      <value>MM/dd/yyyy</value>
    </value-param>

    <value-param>
      <name>timeFormat</name>
      <value>HH:mm</value> <!-- HH:mm/hh:mm a -->
    </value-param>

    <value-param>
      <name>localeId</name>
      <value>BEL</value>
    </value-param>

    <value-param>
      <name>timezoneId</name>
      <value>Europe/Brussels</value>
    </value-param>

    <value-param>
      <name>baseUrlForRss</name>
      <value/>
    </value-param>

    <value-param>
      <name>isShowWorkingTime</name>
```

```
      <value>false</value><!-- boolean true/false -->
    </value-param>

    <value-param>
      <name>workingTimeBegin</name>
      <value>08:00</value><!-- -->
    </value-param>

    <value-param>
      <name>workingTimeEnd</name>
      <value>18:00</value><!-- -->
    </value-param>

    <values-param>
      <name>ignoredUsers</name>
      <description>Definition users to ignore create default calendar</description>
      <!-- <value>demo</value> <value>marry</value> -->
    </values-param>
  </init-params>
</component-plugin>
```

Details:

- **Name**: `calendar.new.user.event.listener` - The unique key to avoid duplicate names. Users can change it.
- **Set-method**: `addListenerPlugin` - The function is executed at the target of the component to register NewUserListener.
- **Type**: `org.exoplatform.calendar.service.impl.NewUserListener` - The class is set up to execute the creation of database.
- **Description**: It is a plugin used to create default personal calendars.

See the details about the init-params of the component in the following table:

Value-params	Possible value	Default value	Description
defaultEventCategories	`String (Comma separated list of category names)`	`Meeting,Calls,Clients,Holiday,Anniversary`	Default event categories for users.
defaultCalendarCategory	`String`	`Default`	Name of the calendar group.
viewType	`0-6 (see below)`	`1`	Default view after user logs in and goes to the Calendar portlet.
timeInterval	`integer in minutes`	`15`	The time unit interval when you drag and move the event (in Day and Week views only).
weekStartOn	`1-7 (see below)`	`2`	Day to use as the beginning of the week. It only affects the Week view.
dateFormat	`valid Java Date format`	`MM/dd/yyyy`	The display format for dates.

Value-params	Possible value	Default value	Description
timeFormat	valid Java Date format	HH:mm	The display format for time.
localeId	valid locale ID	BEL	Id of the geographic locale.
timezoneIds	valid TimeZone id	Europe	User time zone.
baseUrlForRss	none	none	The URL to publish the RSS content.
isShowWorkingTime	true/false	false	Indicate if the working time should be highlighted in the Day view.
workingTimeBegin	time in timeFormat	08:00	The start time in working time.
workingTimeEnd	time in timeFormat	18:00	The end time in working time.
ignoredUsers	user id, use multiple by each line	demo/marry	Definition users to ignore creating the default calendar.

The **viewType** parameter is encoded by a number as follow:

0 : Day view

1 : Week view

2 : Month view

3 : Year view

4 : List view

5 : Schedule view

6 : Working days view

The **weekStartOn** parameter is encoded as follow:

1 : Sunday

2 : Monday

3 : Tuesday

4 : Wednesday

5 : Thursday

6 : Friday

7 : Saturday

3.2.1.2. NewGroupListener

To use the plugin in the component configuration, you must use the target-component:

```
<target-component>org.exoplatform.services.organization.OrganizationService</target-component>
```

The configuration is applied mainly in *extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/calendar/calendar-service-configuration.xml*.

```
<component-plugin>
  <name>calendar.new.group.event.listener</name>
  <set-method>addListenerPlugin</set-method>
  <type>org.exoplatform.calendar.service.impl.NewGroupListener</type>
  <description>description</description>
  <init-params>
    <value-param>
      <name>defaultEditPermission</name>
      <value>*.*</value><!-- Multi value membership, use coma (,) to split values -->
    </value-param>
    <value-param>
      <name>defaultViewPermission</name>
      <value>*.*</value>
    </value-param>
    <value-param>
      <name>defaultLocale</name>
      <value>BEL</value>
    </value-param>
    <value-param>
      <name>defaultTimeZone</name>
      <value>Europe/Brussels</value>
    </value-param>
    <values-param>
      <name>ignoredGroups</name>
      <description>Definition group to ignore create default calendar</description>
      <!-- <value>/platform/guests</value> -->
      <value>/spaces/*</value> <!-- single value, use more <value> tags to add more group -->
    </values-param>
  </init-params>
</component-plugin>
```

Details:

- **Name**: `calendar.new.group.event.listener` - The unique key to avoid duplicate names. Users can change it.
- **Set-method**: `addListenerPlugin` - The function is executed at the target of the component to register NewGroupListener.
- **Type**: `org.exoplatform.calendar.service.impl.NewGroupListener` - The class which is set up to execute the creation of database.
- **Description** - It is the plugin used to create default group calendars.

See the details about the init-params of the component in the following table:

Value-params	Possible values	Default values	Description
defaultEditPermission	`User id` (Multi value membership, use coma (,)	`.` means that all members in that group	The default permission assigned to membership

Value-params	Possible values	Default values	Description
	to split values)	can modify and add, remove a calendar, events/tasks of the calendar	in a specific group to edit calendars and events/tasks of the calendar.
defaultViewPermission	User id (Multi value membership, use coma (,) to split values)	. means that all members in that group can view this calendar and all the events/tasks of this calendar.	The default permission assigned to membership in a specific group to view a calendar and events /tasks of the calendar.
defaultLocale	Valid locade id	BEL (see more locale ids http://userpage.chemie.fu-berlin.de/diverse/doc/ISO_3166)	The default locale of the calendar.
defaultTimeZone	Valid timezone id	Europe/Brussels (see more for timeZone ids http://www.unicode.org/cldr/data/docs/design/formatting/	The default time zone of the calendar.
ignoredGroups	Group id (use line to define multiple value)	/platform/guests	Definition group to ignore create the default calendar.

3.2.1.3. NewMembershipListener

To use the plugin in the component configuration, you must use the target-component:

```
<target-component>org.exoplatform.services.organization.OrganizationService</target-component>
```

The configuration is applied mainly in *extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/calendar/calendar-service-configuration.xml*.

```
<component-plugin>
  <name>calendar.new.membership.event.listener</name>
  <set-method>addListenerPlugin</set-method>
  <type>org.exoplatform.calendar.service.impl.NewMembershipListener</type>
  <description>description</description>
</component-plugin>
```

Details:

- **Name**: `calendar.new.membership.event.listener` - The unique key to avoid duplicate names. Users can change it.
- **Set-method**: `addListenerPlugin` - The function which is executed at the target of the component.
- **Type**: `org.exoplatform.calendar.service.impl.NewMembershipListener` - The class which is set up to execute the creation of database.

- **Description**: It is a plugin used to execute sending reminder emails to users.

3.2.1.4. ReminderPeriodJob

The Calendar application of eXo Collaboration can send event reminders by using the email reminder plugin configuration. You will probably need to adjust this configuration to meet your own needs. The feature is based on a periodic poll of the stored reminders.

You must use the following target component to use the plugin in this configuration:

```
<target-component>org.exoplatform.services.scheduler.JobSchedulerService</target-component>
```

The configuration is applied in *extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/cs-configuration.xml*.

```
<component-plugin>
  <name>RecordsJob</name>
  <set-method>addPeriodJob</set-method>
  <type>org.exoplatform.calendar.service.ReminderPeriodJob</type>
  <description>add e-mail reminder job to the JobSchedulerService</description>
  <init-params>
    <properties-param>
      <name>job.info</name>
      <description>save the monitor data periodically</description>
      <property name="jobName" value="ReminderJob"/>
      <property name="groupName" value="CollaborationSuite"/>
      <property name="job" value="org.exoplatform.calendar.service.ReminderJob"/>
      <property name="repeatCount" value="0"/>
      <property name="period" value="180000"/>
      <property name="startTime" value="+0"/>
      <property name="endTime" value=""/>
    </properties-param>
  </init-params>
</component-plugin>
```

Details:

- **Name**: `RecordsJob` - The name of a schedule job.
- **Set-method**: `addPeriodJob` - The plugin which registers the method.
- **Type**: `org.exoplatform.calendar.service.ReminderPeriodJob` - A class that executes to transfer data into database of Job Scheduler.
- **Description**: Add email reminder job to the JobSchedulerService.

See details about the init-params of the component in the following table:

Properties-param	Description	Property names	Description	Possible values	Default values
job.info	Save the monitor data periodically.	`jobName`	The name of job	`String`	`ReminderJob`
		`groupName`	The name of group job.	`String`	`CollaborationSuite`

Properties-param	Description	Property names	Description	Possible values	Default values
		job	The name of actual job class.	Class path	org.exoplatform.calendar
		repeatCount	How many times to run this job.	Long	0, (use '0' which means 'run forever)
		period	The time interval (millisecond) between job executions.	Long	180000
		startTime	The time when the job starts running.	Integer	+0
		endTime	The time when the job ends running.	Integer	none

3.2.1.5. PopupReminderPeriodJob

You must use the following target component to use the plugin in this configuration:

```
<target-component>org.exoplatform.services.scheduler.JobSchedulerService</target-component>
```

The configuration is applied in *extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/cs-configuration.xml*.

```
<component-plugin>
  <name>PopupRecordsJob</name>
  <set-method>addPeriodJob</set-method>
  <type>org.exoplatform.calendar.service.PopupReminderPeriodJob</type>
  <description>add popup reminder job to the JobSchedulerService</description>
  <init-params>
    <properties-param>
      <name>job.info</name>
      <description>save the monitor data periodically</description>
      <property name="jobName" value="PopupReminderJob"/>
      <property name="groupName" value="CollaborationSuite"/>
      <property name="job" value="org.exoplatform.calendar.service.PopupReminderJob"/>
      <property name="repeatCount" value="0"/>
      <property name="period" value="6000"/>
      <property name="startTime" value="+0"/>
      <property name="endTime" value=""/>
    </properties-param>
    <properties-param>
      <name>popupreminder.info</name>
      <description>save the monitor data periodically</description>
      <property name="portalName" value="portal"/>
    </properties-param>
  </init-params>
</component-plugin>
```

Details:

- **Name**: `PopupRecordsJob` - The name of the job.
- **Set-method**: `addPeriodJob` - The plugin registering method.
- **Type**: `org.exoplatform.calendar.service.PopupReminderPeriodJob` - The class which executes to transfer the data into database of Job Scheduler.
- **Description**: Add popup reminder job to the JobSchedulerService.

See details about the init-params of the component in the following table:

- Properties-param: **job.info**. This param saves the monitor data periodically and includes the following sub-params:

Properties-param	**Description**	**Property names**	**Description**	**Possible values**	**Default values**	
job.info	Save the monitor data periodically.	`jobName`	The name of job.	`String`	`PopupReminderJob`	
		`groupName`	The name of group job.	`String`	`CollaborationSuite`	
		`job`	The name of actual job class.	`Class path`	`org.exoplatform.calendar.service.Pop`	
		`repeatCount`	How many times to run this job.	`Long`	`0, ( use '0' which means 'run forever)`	
		`period`	The time interval (millisecond) between job executions.	`Long`	`6000`	
		`startTime`	The time when the job starts running.	`Long`	`+0`	
		`endTime`	The time when the job ends running.	`Integer`	`None`	
popupreminderinfo	Save the monitor data periodically.	`portalName`	The name of the portal.	`String`	`portal`	##

3.2.2. AddActionsPlugin

The configuration of the AddActionsPlugin is found in *WEB-INF/cs-extension/cs/webservice/webservice-configuration.xml*.

It is used to register the listener named `org.exoplatform.webservice.cs.LastUpdateAction` and it is executed, basing on eventTypes.

```
<component>
  <type>org.exoplatform.services.jcr.impl.ext.action.SessionActionCatalog</type>
  <component-plugins>
    <component-plugin>
      <name>Last Update Action</name>
      <set-method>addPlugin</set-method>
      <type>org.exoplatform.services.jcr.impl.ext.action.AddActionsPlugin</type>
      <description>add actions plugin</description>
      <init-params>
        <object-param>
          <name>actions</name>
          <object type="org.exoplatform.services.jcr.impl.ext.action.AddActionsPlugin$ActionsConfig">
            <field name="actions">
              <collection type="java.util.ArrayList">
                <value>
                  <object type="org.exoplatform.services.jcr.impl.ext.action.ActionConfiguration">
                    <field name="eventTypes">
                      <string>addNode,changeProperty</string>
                    </field>
                    <field name="nodeTypes">
                      <string>exo:calendarEvent</string>
                    </field>
                    <field name="actionClassName">
                      <string>org.exoplatform.webservice.cs.LastUpdateAction</string>
                    </field>
                  </object>
                </value>
              </collection>
            </field>
          </object>
        </object-param>
      </init-params>
    </component-plugin>
  </component-plugins>
</component>
```

- **object-param**: `Actions` - The name of the object.
- **object type**: `org.exoplatform.services.jcr.impl.ext.action.AddActionsPlugin$ActionsConfig` - A class used to register the following field names in the table below:

Field name	String	Description
eventTypes	`addNode,changeProperty`	The type of the event.
nodeTypes	`exo:calendarEvent`	The type of the node.
actionClassName	`org.exoplatform.webservice.cs.LastUpdateAction`	The registration class to execute the actions that the plugin requires.

3.2.3. Chat Configuration

The Chat congiguration is applied in */extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/chat/chat-service-configuration.xml*.

3.2.3.1. HistoryPeriodJob

```
<external-component-plugins>
  <target-component>org.exoplatform.services.scheduler.JobSchedulerService</target-component>
  <component-plugin>
    <name>ChatRecordsJob</name>
    <set-method>addPeriodJob</set-method>
    <type>org.exoplatform.services.xmpp.connection.impl.HistoryPeriodJob</type>
    <description>add chat messages from Openfire Server to History</description>
    <init-params>
      <properties-param>
        <name>job.info</name>
        <description>save the monitor data periodically</description>
        <property name="jobName" value="messageToHistoricalMessageJob"/>
        <property name="groupName" value="CollaborationSuite"/>
        <property name="job" value="org.exoplatform.services.xmpp.connection.impl.HistoryJob"/>
        <property name="repeatCount" value="0"/>
        <property name="period" value="3000"/>
        <property name="startTime" value="+0"/>
        <property name="endTime" value=""/>
      </properties-param>
      <properties-param>
        <name>history.info</name>
        <description>save the monitor data periodically</description>
        <property name="logBatchSize" value="50"/>
      </properties-param>
    </init-params>
  </component-plugin>
</external-component-plugins>
```

Details:

- **Name**: `ChatRecordsJob` - The name of the job.
- **Set-method**: `addPeriodJob` - The plugin which registers the method.
- **Type**: `org.exoplatform.services.xmpp.connection.impl.HistoryPeriodJob` - A class which executes to transfer the data into the database of Job Scheduler.
- **Description**: It is used to save chat messages from Openfire Server to History.

See details about the init-params of the component in the following table:

Properties-param	Description	Property names	Description	Possible values	Default values
job.info	Save the monitor data periodically.	`jobName`	The name of job.	`String`	`messageToHistoricalMessag`
		`groupName`	The name of group name.	`String`	`CollaborationSuite`
		`job`	The name of actual job class.	`Class path`	`org.exoplatform.services.`
		`repeatCount`	How many times to run this job.	`integer`	`0` (use '0' which means 'run forever)
		`period`	The time interval	`Long`	`3000`

Properties-param	Description	Property names	Description	Possible values	Default values
			(millisecond) between job executions.		
		`startTime`	The time the job starts running.	`Long`	`+0`
		`endTime`	The time the job ends running.	`Long`	`none`
history.info	Save the monitor data periodically.	`logBatchSize`	The maximum number of messages in the cache are saved once the job runs.	`Integer`	`50`

3.2.3.2. RequestFilterComponentPlugin

```
<external-component-plugins>
  <target-component>org.exoplatform.services.rest.impl.RequestHandlerImpl</target-component>
  <component-plugin>
    <name>ws.rs.request.filter</name>
    <set-method>addPlugin</set-method>
    <type>org.exoplatform.services.rest.impl.RequestFilterComponentPlugin</type>
    <init-params>
      <value-param>
        <name>RESTXMPPServiceFilter</name>
        <value>org.exoplatform.services.xmpp.rest.filter.RESTXMPPServiceFilter</value>
      </value-param>
    </init-params>
  </component-plugin>
</external-component-plugins>
```

Details:

- **Name**: `ws.rs.request.filter` - The name of the filter.
- **Set-method**: `addPlugin` - The plugin which registers the method.
- **Type**: `org.exoplatform.services.rest.impl.RequestFilterComponentPlugin` - The class which executes the requests of the plugin.
- **Description**: It is used to delete the session of a user when he suddenly closes the browser or changes the session.

See details about the init-params of the component in the following table:

Value-param	Description	Possible value	Default value
RESTXMPPServiceFilter	The name of the filter.	`Class path`	`org.exoplatform.services.xmpp.re`

3.2.3.3. AuthenticationLoginListener and AuthenticationLogoutListener

Two functions, including login and logout of XMPPRestService, are responsible for creating a new XMPPSessionImpl and destroying an existing XMPPSessionImpl. They can be called by listeners: AuthenticationLoginListener, AuthenticationLogoutListener or from client(browser) through the REST protocol (jabberLogin, jabberLogout in UIMainChatWindow.js). You must use the same target component for two external component plugins:

```
<target-component>org.exoplatform.services.listener.ListenerService</target-component>
```

3.2.3.3.1. AuthenticationLoginListener

```
<component-plugin>
  <name>exo.core.security.ConversationRegistry.register</name>
  <set-method>addListener</set-method>
  <type>org.exoplatform.services.xmpp.connection.impl.AuthenticationLoginListener</type>
  <description>description</description>
</component-plugin>
```

Details:

- **Name**: `exo.core.security.ConversationRegistry.register` - The name of plugin.
- **Set-method**: `addListener` - The plugin which registers the method.
- **Type**: `org.exoplatform.services.xmpp.connection.impl.AuthenticationLoginListener` - The class to execute the requests of the plugin.
- **Description**: It is used to start the session and log in the chat server.

3.2.3.3.2. AuthenticationLogoutListener

```
<component-plugin>
  <name>exo.core.security.ConversationRegistry.unregister</name>
  <set-method>addListener</set-method>
  <type>org.exoplatform.services.xmpp.connection.impl.AuthenticationLogoutListener</type>
  <description>description</description>
</component-plugin>
```

Details:

- **Name**: `exo.core.security.ConversationRegistry.register` - The name of plugin.
- **Set-method**: `addListener` - The plugin which registers the method.
- **Type**: `org.exoplatform.services.xmpp.connection.impl.AuthenticationLogoutListener` - A class to execute the requests of the plugin.
- **Description**: It is used to end the session and log in the chat server.

3.2.4. Contact Configuration

The Contact Application is configured by three external component plugins: `NewUserListener`, `NewMembershipListener` and `UpdateUserProfileListener`. They use the same target component:

```
<target-component>org.exoplatform.services.organization.OrganizationService</target-component>
```

The Contact configuration is found in *extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/contact/contact-service-configuration.xml*.

3.2.4.1. NewUserListener

```
<component-plugin>
  <name>contact.new.user.event.listener</name>
  <set-method>addListenerPlugin</set-method>
  <type>org.exoplatform.contact.service.impl.NewUserListener</type>
  <description>description</description>
</component-plugin>
```

Details:

- **Name**: `contact.new.user.event.listener` - The name of listener.
- **Set-method**: `addListenerPlugin` - The plugin which registers the method.
- **Type**: `org.exoplatform.contact.service.impl.NewUserListener` - A class that executes all requirements of the plugin.
- **Description**: It is used to initialize personal contact data for users.

3.2.4.2. NewMembershipListener

```
<component-plugin>
  <name>contact.new.membership.event.listener</name>
  <set-method>addListenerPlugin</set-method>
  <type>org.exoplatform.contact.service.impl.NewMembershipListener</type>
  <description>description</description>
</component-plugin>
```

Details:

- **Name**: `contact.new.membership.event.listener` - The name of the listener.
- **Set-method**: `addListenerPlugin` - The plugin which registers the method. Keep this value.
- **Type**: `org.exoplatform.contact.service.impl.NewMembershipListener` - The class which executes all requirements of the plugin.
- **Description**: It is used to initialize an address book for a specific group.

3.2.4.3. UpdateUserProfileListener

```
<component-plugin>
  <name>contact.new.userprofile.event.listener</name>
```

```
  <set-method>addListenerPlugin</set-method>
  <type>org.exoplatform.contact.service.impl.UpdateUserProfileListener</type>
  <description>description</description>
</component-plugin>
```

Details:

- **Name**: `contact.new.userprofile.event.listener` - The name of the listener.
- **Set-method**: `addListenerPlugin` - The plugin which registers the method. Keep this value.
- **Type**: `org.exoplatform.contact.service.impl.UpdateUserProfileListener` - The class which executes all the requirements of the plugin.
- **Description**: It is used to update the personal profile of a user when he changes it on the portal.

3.2.5. Content Configuration

The Content application, such as RSS reader of eXo Collaboration, is configured by two external component plugins: RSSContentPlugin and DescriptionPlugin. Both the external components plugins use the same target component:

```
<target-component>org.exoplatform.content.service.ContentDAO</target-component>
```

This content configuration is applied in *extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/content/content-service-configuration.xml*.

3.2.5.1. RSSContentPluginDescriptionPlugin

```
<component-plugin>
  <name>rssreader.listener</name>
  <set-method>addPlugin</set-method>
  <type>org.exoplatform.content.service.RSSContentPlugin</type>
  <description>rss reader plugin</description>
</component-plugin>
```

Details:

- **Name**: `rssreader.listener` - The name of the listener.
- **Set-method**: `addPlugin` - The plugin which registers the method. Keep this value.
- **Type**: `org.exoplatform.content.service.RSSContentPlugin` - A class which extends ContentPlugin and implements the loadContentMeta method to get content items.
- **Description**: It is a formater used to analyze the data from a RSS resource.

3.2.5.2. DescriptionPlugin

```
<component-plugin>
  <name>description.listener</name>
  <set-method>addPlugin</set-method>
  <type>org.exoplatform.content.service.DescriptionPlugin</type>
```

```
  <description>Description plugin</description>
</component-plugin>
```

Details:

- **Name**: `description.listener` - The name of the listener.
- **Set-method**: `addPlugin` - The plugin registering method. Keep this value.
- **Type**: `org.exoplatform.content.service.DescriptionPlugin` - A class which executes all requirements of the plugin.
- **Description**: It is a plugin to represent data from a RSS source.

3.2.6. Mail Configuration

3.2.6.1. AuthenticationLogoutListener

In the Mail application of eXo Collaboration, when a user checks messages for one account, the remote mailbox fetch is performed as a background job. Before eXo Collaboration 1.2, the job was continued until all messages had been retrieved or when the user stopped the check through the UI. Hence, even when a user was not logged in, the background job was continued. This can be the resource intensive for the server if many users have large mailboxes. Since eXo Collaboration 1.2, one capability is added to halt the background job when the user session terminates (logout or timeout). It makes eXo Collaboration more friendly with server resources. If you want to activate this feature, you need to add a bunch of the xml congifuration in */extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/mail/mail-service-configuration.xml*:

```
<external-component-plugins>
  <target-component>org.exoplatform.services.listener.ListenerService</target-component>
   <component-plugin>
    <name>exo.core.security.ConversationRegistry.unregister</name>
    <set-method>addListener</set-method>
    <type>org.exoplatform.mail.service.AuthenticationLogoutListener</type>
    <description>description</description>
  </component-plugin>
</external-component-plugins>
```

Details:

- **Name**: `exo.core.security.ConversationRegistry.unregister` - The name of listener.
- **Set-method**: `addListener` The plugin which registers the method. Keep this value.
- **Type**: `org.exoplatform.mail.service.AuthenticationLogoutListener` - A class which executes all requirements of the plugin.
- **Description**: It is a plugin used to stop checking mails of a user when he logs out.

3.2.6.2. MailSettingConfigPlugin

Since eXo Collaboration 2.2.0, *MailSettingConfigPlugin* is used to define the behavior, for example, showing/hiding fields and checking/unchecking checkboxes, of email account settings in the mail-server-configuration.xml file. It allows administrators to preconfigure all settings and to specify if end-users have the modification right on each specific setting or not.

```
<external-component-plugins>
  <target-component>org.exoplatform.mail.service.MailService</target-component>
        <component-plugin>
                <name>cs.mail.service.settings</name>
            <set-method>addPlugin</set-method>
                <type>org.exoplatform.mail.service.MailSettingConfigPlugin</type>
                <description>description</description>
                        <init-params>
                                <object-param>
                                        <name>leaveOnServer</name>
                                   <description>options to keep a copy of the message on the mail server after e
                                   <object type="org.exoplatform.mail.service.MailSettingConfig">
                                     <field name="name"><string>leaveOnServer</string></field>
                                     <field name="userAllowed"><boolean>true</boolean></field>
                                     <field name="defaultValue"><string>true</string></field>
                                   </object>
                                </object-param>
                                <object-param>
                                        <name>incomingServer</name>
                                   <description>default incoming server to check for new mails.</description>
                                   <object type="org.exoplatform.mail.service.MailSettingConfig">
                                     <field name="name"><string>incomingServer</string></field>
                                     <field name="userAllowed"><boolean>true</boolean></field>
                                     <field name="defaultValue"><string>imap.gmail.com</string></field>
                                   </object>
                                </object-param>
                                <object-param>
                                        <name>incomingPort</name>
                                   <description>default port incoming server to check for new mails.</descriptio
                                   <object type="org.exoplatform.mail.service.MailSettingConfig">
                                     <field name="name"><string>incomingPort</string></field>
                                     <field name="userAllowed"><boolean>true</boolean></field>
                                     <field name="defaultValue"><string>993</string></field>
                                   </object>
                                </object-param>
                                <object-param>
                                        <name>outgoingServer</name>
                                   <description>description</description>
                                   <object type="org.exoplatform.mail.service.MailSettingConfig">
                                     <field name="name"><string>outgoingServer</string></field>
                                     <field name="userAllowed"><boolean>true</boolean></field>
                                     <field name="defaultValue"><string>smtp.gmail.com</string></field>
                                   </object>
                                </object-param>
                                <object-param>
                                        <name>outgoingPort</name>
                                   <description>description</description>
                                   <object type="org.exoplatform.mail.service.MailSettingConfig">
                                     <field name="name"><string>outgoingPort</string></field>
                                     <field name="userAllowed"><boolean>true</boolean></field>
                                     <field name="defaultValue"><string>465</string></field>
                                   </object>
                                </object-param>
                                <object-param>
                                        <name>acceptIncomingSecureAuthentication</name>
                                   <description>description</description>
                                   <object type="org.exoplatform.mail.service.MailSettingConfig">
                                     <field name="name"><string>acceptIncomingSecureAuthentication</string></fie
                                     <field name="userAllowed"><boolean>true</boolean></field>
                                     <field name="defaultValue"><string>true</string></field>
                                   </object>
                                </object-param>
                                <object-param>
                                        <name>incomingSecureAuthentication</name>
                                   <description>description</description>
                                   <object type="org.exoplatform.mail.service.MailSettingConfig">
                                     <field name="name"><string>incomingSecureAuthentication</string></field>
                                     <field name="userAllowed"><boolean>true</boolean></field>
                                     <field name="defaultValue"><string>starttls</string></field>
                                   </object>
                                </object-param>
                                <object-param>
                                        <name>incomingAuthenticationMechanism</name>
                                   <description>description</description>
                                   <object type="org.exoplatform.mail.service.MailSettingConfig">
                                     <field name="name"><string>incomingAuthenticationMechanism</string></field
                                     <field name="userAllowed"><boolean>true</boolean></field>
```

```
                                    <field name="defaultValue"><string>plain</string></field>
                                 </object>
                              </object-param>
                              <object-param>
                                    <name>acceptOutgoingSecureAuthentication</name>
                                 <description>description</description>
                                 <object type="org.exoplatform.mail.service.MailSettingConfig">
                                    <field name="name"><string>acceptOutgoingSecureAuthentication</string></fie
                                    <field name="userAllowed"><boolean>true</boolean></field>
                                    <field name="defaultValue"><string>true</string></field>
                                 </object>
                              </object-param>
                              <object-param>
                                    <name>outgoingSecureAuthentication</name>
                                 <description>description</description>
                                 <object type="org.exoplatform.mail.service.MailSettingConfig">
                                    <field name="name"><string>outgoingSecureAuthentication</string></field>
                                    <field name="userAllowed"><boolean>true</boolean></field>
                                    <field name="defaultValue"><string>starttls</string></field>
                                 </object>
                              </object-param>
                              <object-param>
                                    <name>outgoingAuthenticationMechanism</name>
                                 <description>description</description>
                                 <object type="org.exoplatform.mail.service.MailSettingConfig">
                                    <field name="name"><string>outgoingAuthenticationMechanism</string></field>
                                    <field name="userAllowed"><boolean>true</boolean></field>
                                    <field name="defaultValue"><string>plain</string></field>
                                 </object>
                              </object-param>
                           </init-params>
        </component-plugin>
 </external-component-plugins>
```

Details:

- **Name**: `cs.mail.service.settings` - The name of the mail settings.
- **Set-method**: `addPlugin` - The plugin which registers the method.
- **Type**: `org.exoplatform.mail.service.MailSettingConfigPlugin` - The plugin's class name.
- **object type**: `org.exoplatform.mail.service.MailSettingConfig` - A class of object that contains a specific setting.
- **Description**: Describes what the setting is or what the setting does.

Object-param	**Desciption**
leaveOnServer	Options to keep the message on the mail server after it has been downloaded to the Mail application of eXo Collaboration.
incomingServer	The default incoming server used to check new mails.
incomingPort	The default port of the incoming server used to check new mails.
outgoingServer	The default port of the outgoing server used to send new mails.
outgoingPort	The default outgoing port to send new mails.
acceptIncomingSecureAuthentication	Accept the secure authentication of the incoming server.

Object-param	Desciption
incomingSecureAuthentication	The type of incoming secure authentication.
incomingAuthenticationMechanism	The type of incoming authentication mechanism.
acceptOutgoingSecureAuthentication	Accepts the secure authentication of the outgoing server.
outgoingSecureAuthentication	The type of outgoing secure authentication.
outgoingAuthenticationMechanism	The type of incoming authentication mechanism.

The object parameters have the same field names, but the values of the parameters are different.

Field names	Description	Possible values
name	The field name in the account settings form.	String
userAllowed	Allow users to edit the field in the account settings form or not.	Boolean
defaultValue	The default value of the field in the account settings form.	String

3.2.7. Social Integration Configuration

The Social Integration Configuration is applied in */extension/webapp/src/main/webapp/WEB-INF/cs-extension/cs/social-integration/social-integration-configuration.xml*.

3.2.7.1. CalendarDataInitialize

```
<external-component-plugins>
  <target-component>org.exoplatform.social.core.space.spi.SpaceService</target-component>
  <component-plugin>
    <name>CalendarDataInitialize</name>
    <set-method>addSpaceListener</set-method>
    <type>org.exoplatform.cs.ext.impl.CalendarDataInitialize</type>
    <init-params>
      <value-param>
        <name>portletName</name>
        <value>CalendarPortlet</value>
      </value-param>
    </init-params>
  </component-plugin>
</external-component-plugins>
```

Details:

- **Name**: `CalendarDataInitialize` - The name of plugin.
- **Set-method**: `addSpaceListener` - The plugin which registers the method.
- **Type**: `org.exoplatform.cs.ext.impl.CalendarDataInitialize` - A class that executes all requirements of the plugin.

- **Description**: It is used to initialize a calendar for a group in a specific space.

See the details about the init-params of the component in the following table:

value-param	Description	Possible value	Default value
portletName	The name of the portlet	`String`	`CalendarPortlet`

3.2.7.2. ContactDataInitialize

```
<external-component-plugins>
  <target-component>org.exoplatform.social.core.space.spi.SpaceService</target-component>
  <component-plugin>
    <name>ContactDataInitialize</name>
    <set-method>addSpaceListener</set-method>
    <type>org.exoplatform.cs.ext.impl.ContactDataInitialize</type>
    <init-params>
      <value-param>
        <name>portletName</name>
        <value>ContactPortlet</value>
      </value-param>
    </init-params>
  </component-plugin>
</external-component-plugins>
```

Details:

- **Name**: `ContactDataInitialize` - The name of the plugin.
- **Set-method**: `addSpaceListener` - The plugin which registers the method.
- **Type**: `org.exoplatform.cs.ext.impl.ContactDataInitialize` - A class that executes all the requires of the plugin.
- **Description**: It is a plugin used to initialize an address book for a group in a specific space.

See the details about the init-params of the component in the following table:

value-param	Possible value	Default value	Description
portletName	`String`	`ContactPortlet`	The name of the portlet.

3.2.7.3. ContactSpaceActivityPublisher

```
<external-component-plugins>
  <target-component>org.exoplatform.contact.service.ContactService</target-component>
  <component-plugin>
    <name>ContactEventListener</name>
    <set-method>addListenerPlugin</set-method>
    <type>org.exoplatform.cs.ext.impl.ContactSpaceActivityPublisher</type>
  </component-plugin>
</external-component-plugins>
```

Details:

- **Name**: `ContactEventListener` - The name of the plugin.
- **Set-method**: `addListenerPlugin` - The plugin which registers the method.
- **Type**: `org.exoplatform.cs.ext.impl.ContactSpaceActivityPublisher` - A class that executes all requirements of the plugin.
- **Description**: It is a plugin used to customize the activity status of a specific space when an event happens on an address book.

3.2.7.4. CalendarSpaceActivityPublisher

```
<external-component-plugins>
  <target-component>org.exoplatform.calendar.service.CalendarService</target-component>
  <component-plugin>
    <name>CalendarEventListener</name>
    <set-method>addEventListenerPlugin</set-method>
    <type>org.exoplatform.cs.ext.impl.CalendarSpaceActivityPublisher</type>
  </component-plugin>
</external-component-plugins>
```

Details:

- **Name**: `CalendarEventListener` - The name of the plugin.
- **Set-method**: `addEventListenerPlugin` - The plugin which registers the method.
- **Type**: `org.exoplatform.cs.ext.impl.CalendarSpaceActivityPublisher` - A class that executes all the requires of the plugin.
- **Description**: It is a plugin used to customize the activity status of a specific space when an event happens on a calendar.

3.2.7.5. 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>CalendarPortlet</value>
        <value>ContactPortlet</value>
      </values-param>
    </init-params>
  </component-plugin>
</external-component-plugins>
```

Details:

- **Name**: `ortlets.prefs.required` - The name of the plugin.
- **Set-method**: `setPortletsPrefsRequired` - The plugin which registers the method.
- **Type**: `org.exoplatform.social.core.application.PortletPreferenceRequiredPlugin` - A class that

executes all the requires of the plugin.

- **Description**: It is a plugin used to declare the application that will automatically create database.

See the details about the init-params of the component in the following table:

value-param	Possible value	Default value	Description
portletsPrefsRequired	`String`	`ContactPortlet / ContactPortlet`	The name of plugin added to SpaceService.

3.3. Data Injectors

Data injectors are used to create data for performance test. This part will describe which data injectors are implemented in eXo Collaboration and how to use them.

In eXo Collaboration, data injectors are implemented as plug-ins attached to the *org.exoplatform.services.bench.DataInjectorService* service and handled via RESTful requests. This service is normally registered to the portal container as a general component.

To use this service, add the following dependency to the Classpath of the server:

```
<dependency>
      <groupId>org.exoplatform.commons</groupId>
      <artifactId>exo.platform.commons.component</artifactId>
</dependency>
```

To activate the *DataInjectorService* component, you need to register it to a portal container by the following configuration:

```
<component>
    <type>org.exoplatform.services.bench.DataInjectorService</type>
</component>
```

When you want to inject data for a specific product, you must implement a class which extends "org.exoplatform.services.bench.DataInjector" and register it to the *DataInjectorService* component as a plug-in.

In eXo Collaboration, there are three plug-ins attached to the *DataInjectorService* component:

- ContactDataInjector
- CalendarDataInjector
- MailDataInjector

3.3.1. ContactDataInjector

The *ContactDataInjector* plug-in is used to manage data injection of the Address Book application.

To use this plug-in, do as follows:

1. Add the following configuration to the *configuration.xml* file to register the plug-in to the *DataInjectorService* component:

```
<external-component-plugins>
    <target-component>org.exoplatform.services.bench.DataInjectorService</target-component>
    <component-plugin>
      <name>ContactDataInjector</name>
      <set-method>addInjector</set-method>
      <type>org.exoplatform.contact.service.bench.ContactDataInjector</type>
      <description>inject data for Contact</description>
      <init-params>
        <value-param>
          <name>mA</name> <!-- maximum number of address books -->
          <value>5</value>
        </value-param>
        <value-param>
          <name>mC</name> <!-- maximum number of contacts per a address books -->
          <value>10</value>
        </value-param>
      </init-params>
    </component-plugin>
</external-component-plugins>
```

In which:

- **Name**: `ContactDataInjector`
- **Set-method**: `addInjector`
- **Type**: `org.exoplatform.contact.service.bench.ContactDataInjector`

Parameters	Possible Values	Default Values	Description
mA	number	5	The maximum number of address books of the Address Book application in each injection.
mC	number	5	The maximum number of contacts of an address book in each injection.

2. Execute the injector by the RESTful request as follows:

```
http://{domain}/{rest}/bench/inject/ContactDataInjector?mA=5&mC=10
```

In which:

- **{domain}**: The domain name of the server.
- **{rest}**: The name of eXo REST service.

3.3.2. CalendarDataInjector

The *CalendarDataInjector* plug-in is used to manage data injection in the Calendar application.

To use this plug-in, do as follows:

1. Add the following configuration to the *configuration.xml* file to register the plug-in to the *DataInjectorService* component:

```
<external-component-plugins>
    <target-component>org.exoplatform.services.bench.DataInjectorService</target-component>
    <component-plugin>
      <name>CalendarDataInjector</name>
      <set-method>addInjector</set-method>
      <type>org.exoplatform.calendar.bench.CalendarDataInjector</type>
      <description>inject data for Calendar</description>
      <init-params>
        <value-param>
          <name>mCt</name> <!-- maximum number of categories -->
          <value>5</value>
        </value-param>
        <value-param>
          <name>mEcat</name> <!-- maximum number of event categories -->
          <value>10</value>
        </value-param>
        <value-param>
          <name>mCal</name> <!-- maximum number of calendars -->
          <value>10</value>
        </value-param>
        <value-param>
          <name>mEv</name> <!-- maximum number of events -->
          <value>5</value>
        </value-param>
        <value-param>
          <name>mTa</name> <!-- maximum number of tasks -->
          <value>5</value>
        </value-param>
        <value-param>
          <name>typeOfInject</name> <!-- type of inject -->
          <value>all</value> <!-- string all/public/private -->
        </value-param>
      </init-params>
    </component-plugin>
</external-component-plugins>
```

In which:

- **Name**: `CalendarDataInjector`
- **Set-method**: `addInjector`
- **Type**: `org.exoplatform.calendar.bench.CalendarDataInjector`

Parameters	Possible Values	Default Values	Description
mCt	number	5	The maximum number of categories in each injection.
mEcat	number	10	The maximum number of event categories in each injection.

Parameters	Possible Values	Default Values	Description
mCal	number	10	The maximum number of calendars in each injection.
mEv	number	5	The maximum number of events in one calendar in each injection.
mTa	number	5	The maximum number of tasks in one calendar in each injection.
typeOfInject	String	all	The type of injection, including "public", "private" and "all". If the type is set as "public", only public calendars are created. If the type is set as "private", only personal calendars are create. If the type is set as "all", both the public and private ones are created.

2. Execute the injector by the RESTful request as follows.

```
http://{domain}/{rest}/bench/inject/CalendarDataInjector?mCt=5&mEcat=10&mCal=10&mEv=20&mTa=20&typeOfInject=priva
```

3.3.3. MailDataInjector

The *MailDataInjector* is used to manage data injection in the Mail application.

Because of the quite complicated architecture of the Mail application in eXo Collaboration, the injector uses a simple mail server (a mock server) to simulate the way a mail server works. This will create real mail data without mocking effort from the injector and create a reliable testing environment.

To use the injector, do as follows:

1. Add the Greenmail library to the classpath of the web server (in tomcat, copy it to the **lib** folder). The library is Greenmail 1.3.1b but including some bug fixes which are not updated in the counterpart version of Icegreen. (To have the library, contact eXo Support team).

2. Initialize this component as a service of GateIn, then it will be invoked by MailDataInjector.

```
<component>
    <type>org.exoplatform.mail.service.bench.SimpleMailServerInitializer</type>
</component>
```

3. Register *MailDataInjector* to *DataInjectorService* by the following configuration:

```
<external-component-plugins>
  <target-component>org.exoplatform.services.bench.DataInjectorService</target-component>
  <component-plugin>
      <name>MailDataInjector</name>
      <set-method>addInjector</set-method>
      <type>org.exoplatform.mail.service.bench.MailDataInjector</type>
      <description>inject data for Contact</description>
    </component-plugin>
</external-component-plugins>
```

In which:

- **Name**: `MailDataInjector`
- **Set-method**: `addInjector`
- **Type**: `org.exoplatform.mail.service.bench.MailDataInjector`

3.3.4. Usage of MailDataInjector

- To insert mail data into the Mail application, the request link is as follows:

```
http://{domain}/{rest}/bench/inject/MailDataInjector/?users=mary,root&accounts=2,account&inPro=IMAP&check=true&
```

Params	Values	Description
users	String	The list of users sepereated by commas.
accounts	String	The number of accounts injected by the data injector. This value consists of two parts seperated by commas: the first is the number of accounts of an user, the second is the prefix of the account Id.
inPro	String	The incoming protocol. The possible values are: POP3 and IMAP.
check	Boolean	Specify if checking mails after the data are created on the mail server or not. If the value is true, the injector will execute checking new message tasks sequentially. This task can take some time if the data are large.
msgs	Number	The number of messages will be available in each account.
attSize	Number	The size of an attachment. If it does not exist or is equal to 0, no

Params	Values	Description
		file is attached to the mail message.

- To reject mail data from the Mail application, the request link is as follows:

```
http://{domain}/{rest}/bench/reject/MailDataInjector/?users=root&accounts=2,account
```

This link will request for removing 2 accounts of "root" of which Id starts with "account".

Note

By default, such settings have been declared in "csdemo.war/WEB-INF/conf/csdemo/cs/bench-configuration.xml". Therefore, to save time, you should import the *bench-configuration.xml* file to "*csdemo.war/WEB-INF/conf/configuration.xml*", and then modify it as your purpose rather than create a new one.

3.4. eXo Chatserver Configuration

3.4.1. Openfire Configuration

The Chat service of eXo Collaboration is a Jabber engine powered by Openfire. eXo Platform will delegate the actual Jabber protocol communication to Openfire.

You have the full latitude to configure Openfire. There are two possible ways to do it:

- The admin console: http://localhost:9090/.
- The openfire.xml file in *$openfire_home/conf/*.

3.4.1.1. Configuration in Openfire.xml

The Openfire server has a single configuration file called *openfire.xml* and located under the *exo-openfire/conf* directory. Configuration is based on properties expressed in an XML syntax. For example, to set the property prop.name.is.blah=value, you would write this xml snippet:

```
<prop>
  <name>
    <is>
      <blah>value</blah>
    </is>
  </name>
</prop>
```

Openfire has an extensive list of configuration properties. You can read a list of all properties on this page: http://www.igniterealtime.org/community/docs/DOC-1061.

3.4.1.2. eXo specific configuration

The eXo Collaboration bundle comes with a pre-configured Openfire server. It is bundled with some eXo plugins and configurations that allow connectivity with eXo Platform. The key properties for integration are:

- *provider.auth.className*: An implementation of the AuthProvider interface for authentication of users on the Chat server.

- *provider.users.className*: An implementation of the UserProvider interface to which Openfire will delegate users management.

- *provider.groups.className*: An implementation of the GroupProvider interface to which Openfire will delegate groups management.

eXo Platform provides implementations for these 3 interfaces with ExoAuthProvider, ExoUserProvider, ExoGroupProvider. These implementations are based on the eXo REST framework and let you configure the endpoints within the *openfire.xml* file with additional properties:

Property	Default value	Description
eXo.env.serverBaseURL	`http://localhost:8080/##`	The base URL of the server.
eXo.env.restContextName	`rest`	The context name of REST Web application.
provider.authorizedUser.name	`root`	The username to authenticate to access the HTTP REST service.
provider.authorizedUser.password	`gtn`	The password matching with provider.authorizeduser.name.
exoAuthProvider.authenticationURL	`/organization/authenticate/`	The URL to authenticate users.
exoAuthProvider.authenticationMethod	`GET`	The HTTP method used for the authentication method.
exoUserProvider.findUsersURL	`/organization/xml/user/find-all`	The URL to find all users.
exoUserProvider.findUsersMethod	`GET`	The HTTP method used to find all users in the system.
exoUserProvider.getUsersURL	`/organization/xml/user/view-range`	The URL to retrieve a range of users.
exoUserProvider.getUsersMethod	`GET`	The HTTP method used for user/view-range.
exoUserProvider.usersCountURL	`/organization/xml/user/count`	The URL to count the number of users.
exoUserProvider.usersCountMethod	`GET`	The HTTP method used to count the number of users.
exoUserProvider.userInfoURL	`/organization/xml/user/info/`	The URL to get the information of users.
exoUserProvider.userInfoMethod	`GET`	The HTTP method used to get the

Property	Default value	Description
		information of users.
exoGroupProvider.groupInfoURL	`/organization/xml/group/info/`	The URL to get the information of a group of users.
exoGroupProvider.groupInfoMethod	GET	The HTTP method used to get the information of a group of users.
exoGroupProvider.getGroupsAllURL	`/organization/xml/group/view-all/`	The URL to view a list of all user groups.
exoGroupProvider.getGroupsAllMethod	GET	The HTTP method used to view a list of all user groups.
exoGroupProvider.getGroupsRangeURL	`/organization/xml/group/view-from-to/`	The URL to list groups in a specific range.
exoGroupProvider.getGroupsRangeMethod	GET	The HTTP method used to list groups in a specific range.
exoGroupProvider.getGroupsForUserURL	`/organization/xml/group/groups-for-user/`	The URL to list groups to which a user belongs.
exoGroupProvider.getGroupsForUserMethod	GET	The HTTP method used to list groups to which a user belongs.
exoGroupProvider.groupsCountURL	`/organization/xml/group/count`	The URL to count the number of groups.
exoGroupProvider.groupsCountMethod	GET	The HTTP method used to count the number of groups.

As you can see, the default settings will only work if eXo Platform is deployed on the same host as Openfire, on the port 8080.

Note

restContextName is used to specify the Openfire server that is dedicated for the portal. If the *eXo.env.restContextName* system property exists, it will override this value.

The *eXo.env.restContextName* system property can be set by specifying the -D option to the Java command when running Openfire.

For example:

- If the Openfire server is dedicated for the portal named "portal", the command will have the following format: *java -DeXo.env.restContextName=rest -jar ../lib/startup.jar .*
- If the Openfire server is dedicated for the portal named "csdemo", the command will have following format: *java -DeXo.env.restContextName=rest-csdemo -jar ../lib/startup.jar..*

By default, the Openfire server is dedicated to the portal named "portal".

3.4.2. System Configuration

Openfire makes use of several ports for communication.

Interface	Port	Type	Description
All addresses	`5222`	`Client to Server`	The standard port for clients is to connect to the server. Connection may or may not be encrypted. You can update the security settings for this port.
All addresses	`9090 & 9091`	`Admin Console`	The ports used for the unsecured and secured Openfire Admin Console accesses respectively.
All addresses	`7777`	`File Transfer Proxy`	The port used for the proxy service that allows files to be transferred between two entities on the XMPP network.
All addresses	`3478 & 3479`	`STUN Service`	The port used for the service that ensures connectivity between entities behind a NAT.

You can view the table above in *http://hostname:9090/index.jsp* after you have logged into the Openfire's web console and also customize those ports by yourself.

3.4.3. AS configuration

To enable the propagation of identity across the Chat webapp, you are required to enable the SSO valve on the Tomcat-based Application server.

- For the Jboss server, edit *jboss/server/default/deploy/jboss-web.deployer/server.xml*.

- For the Tomcat server, edit *tomcat/conf/server.xml*.

The valve should already be there, you just need to uncomment it if it is not already done.

```
<Valve className="org.apache.catalina.authenticator.SingleSignOn"/>
```

In case of the cluster deployment, you may want to use `ClusteredSingleSignOn` instead.

```
<Valve className="org.jboss.web.tomcat.service.sso.ClusteredSingleSignOn"/>
```

Chapter 4. JCR Structure

eXo Collaboration is a JCR-based product, so data of eXo Collaboration are managed by the eXo-JCR service with each specific structure. The chapter aims at outlining the JCR structure of each application in eXo Collaboration through diagrams and then describing properties of main node types.

Each diagram shows nodes and their primary node types. Every node/child node must have only one primary node type represented in the round bracket () under the node/childnode, but may also have many mixin node types. Because mixin nodes cannot define the node structure like the primary nodes, they are not shown in the diagrams and their properties hereafter are not described.

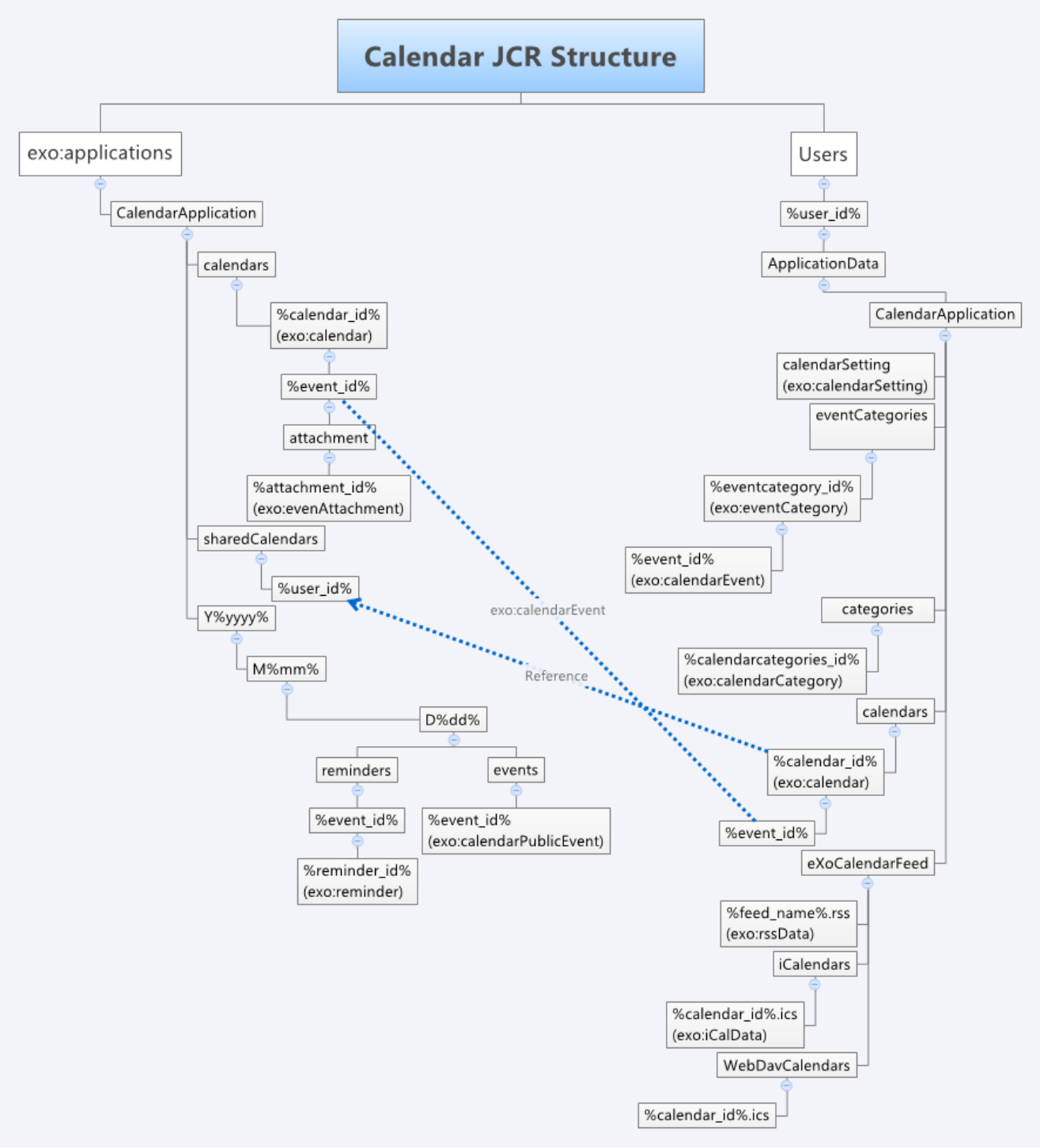
Note

To learn more about the eXo Collaboration JCR Structure, you should have the certain knowledge of JCR.

4.1. Calendar JCR Structure

The Calendar data are saved in eXo-JCR under the CalendarApplication data directory. The Calendar JCR Structure is divided into two main branches: one for public (exo:application) and the other for users (Users).

The whole JCR structure of Calendar can be visualized in the diagram below:



4.1.1. calendars

The **Calendars** node of the nt:unstructured type contains the child nodes of the **exo:calendar** type. When a calendar is created by users or the default ones in the system, it is stored under the **calendars** node: *CalendarApplication/calendars/%calendar_id%*. Its node type is *exo:calendar* that has the following properties:

Property name	Required type	Multiple	Description
exo:id	String	false	The Id of the calendar.
exo:name	String	false	The name of the calendar.
exo:description	String	false	The brief description of

Property name	Required type	Multiple	Description
			the calendar.
exo:viewPermissions	String	true	The list of users/groups having the view permissions.
exo:editPermissions	String	true	The list of users/groups having the edit permissions.
exo:groups	String	true	The list of user groups to which the calendar belongs.
exo:categoryId	String	false	The Id of the category containing the calendar.
exo:calendarColor	String	false	The color name of the calendar that is defined in the *org.exoplatform.web ui.form.ext.UIFormColorPicker class (such as Sky blue, Powder blue).*
exo:calendarOwner	String	false	The name of the user creating the calendar.
exo:locale	String	false	Location where the calendar is set in format of the uppercase ISO 3166 3-letter country code.
exo:timeZone	String	false	The Id of the time zone that is set by the user in compliance with the Java class: java.util.TimeZone.
exo:publicUrl	String	false	The public ICAL link of the calendar.
exo:privateUrl	String	false	The private ICAL link of the calendar.

When a user shares his own calendar with other users, the Id of the calendar node is referred to the node under the **sharedCalendar** node: *CalendarApplication/sharedCalendars/%user_id%* following the JCR reference mechanism.

In case of users' private calendar, two mixin node types *exo:remoteCalendar* and *exo:calendarShared* can be added to the *exo:calendar* node type.

- The *exo:remoteCalendar* mixin node type has the following properties:

Property name	Required type	Multiple	Description
exo:remoteUrl	String	false	The URL of the remote calendar.
exo:remoteType	String	false	The type of the remote calendar, including ICalendar (.ics) and CalDav.
exo:username	String	false	The username used to access the remote calendar.
exo:password	String	false	The password used to access the remote calendar.
exo:syncPeriod	String	false	The period the remote calendar is synchronized. auto, 5 minutes, 10 minutes, 15 minutes, 1 hour, 1 day, 1 year
exo:lastUpdated	Date	false	The last update of the remote calendar.
exo:beforeDate	String	false	The period before the current date in which the calendar is checked out, including the values: None (the unlimited time), 1 week, 2 weeks, 1month, 2 months, 3 months, 6 months and 1 year.
exo:afterDate	String	false	The period after the current date in which the calendar is checked out, including the values: Forever (the unlimited time), 1 week, 2 weeks, 1month, 2 months, 3 months, 6 months and 1 year.

- The *exo:calendarShared* mixin node type has the following properties:

Property name	Required type	Multiple	Description
exo:sharedId	Reference	true	The user Ids who are shared the calendars.

An event can have many attachments which are stored under the **attachment** node of the *exo:eventAttachment*type: *CalendarApplication/calendars/%calendar_id%/%event_id%/attachment/%attachment_id%*. The *exo:eventAttachment* node type has the following properties:

Property name	Required type	Multiple	Description
exo:fileName	String	false	The name of the attached file.

4.1.2. eventCategories

The **eventCategories** node contains all event categories. When an event category is created, it is stored in a node of the *exo:eventCategory* type, under the **eventCategories** node defined at the path: CalendarApplication/eventCategories/%eventcategory_id%.

This node type has the following properties:

Property name	Required type	Multiple	Description
exo:id	String	false	The Id of the category to which an event belongs.
exo:name	String	false	The name of the category to which an event belongs.
exo:description	String	false	The brief description of the category to which an event belongs.

Each event category node contains the calendar event node of the *exo:calendarEvent)* type. This node of the *exo:calendarEvent* type is stored at the path: *CalendarApplication/eventCategories/%eventcategory_id%/%event_id%*.

This node type has the following properties:

Property name	Required type	Multiple	Description
exo:id	String	false	The Id of the event.
exo:eventType	String	false	Type of the event, including Event and Task.
exo:summary	String	false	The summary of the event.
exo:location	String	false	The location where the event will take place.
exo:taskDelegator	String	false	The name of the user being delegated the task.

Property name	Required type	Multiple	Description
exo:description	String	false	The brief description of the event.
exo:eventCategoryId	String	false	The Id of the category containing the event.
exo:eventCategoryName	String	false	The name of the category containing the event.
exo:calendarId	String	false	The Id of the calendar containing the event.
exo:fromDateTime	Date	false	The start time of the event.
exo:toDateTime	Date	false	The end time of the event.
exo:priority	String	false	The preference order of the event, including 4 values: none, low, normal, high.
exo:isPrivate	Boolean	false	Define if the event is private or not.
exo:eventState	String	false	The state of the event which depends on each event type.
exo:invitation	String	true	The list of email addresses of users being invited to the event. This property is for the Event type only.
exo:participant	String	true	The list of users being invited to the event. This property is for the Event type only.
exo:participantStatus	true	String	The status of the participant, including name and status value.
exo:message	String	false	The content of the invitation email.
exo:repeat	String	false	Repetition type of the event, including: "norepeat", "daily", "weekly", "monthly", "yearly", "weekend", "workingdays".
exo:sendOption	String	false	The option to notify users

Property name	Required type	Multiple	Description
			before sending the invitation via email: never (not sending all time), always (sending without asking) and ask (asking before sending).

4.1.3. categories

The **categories** node of the *nt:unstructured* type contains the child nodes of the *exo:calendarCategory* type. These child nodes containing the Id of the calendars in the categories are stored under the **categories** node: *CalendarApplication/categories/%calendarcategories_id%*.

The *exo:calendarCategory* node type has the following properties:

Property name	Required type	Multiple	Description
exo:id	String	false	The Id of the category to which a calendar belongs.
exo:name	String	false	The name of the category to which a calendar belongs.
exo:description	String	false	The brief description of the category to which a calendar belongs.
exo:calendarIds	String	true	A list of calendar Ids belonging to the category.

4.1.4. eXoCalendarFeed

The **eXoCalendarFeed** of the *nt:unstructured* type contains **iCalendars**, **webDavCalendars** as child nodes and others of the *exo:rssData* type.

The *exo:rssData* node type has the following properties:

Property name	Required type	Multiple	Description
exo:baseUrl	String	false	The original link to the RSS source.
exo:title	String	false	The title of the feed.
exo:content	Binary	false	The content of the feed.

The **iCalendars** node of the *nt:unstructured* type contains the child nodes of *exo:iCalData* type.

The *exo:iCalData* node type has the following properties:

Property name	Required type	Multiple	Description
exo:data	Binary	false	The exported content of the calendar in the ics.format.

The **webDavCalendars** node of the *nt:unstructured* type contains the child nodes of the *exo:caldavCalendarEvent* type.

The *exo:caldavCalendarEvent* node type has the following properties:

Property name	Required type	Multiple	Description
exo:caldavHref	String	false	The URL of the remote calendar event.
exo:caldavEtag	String	false	The tag of the remote calendar event.

4.1.5. Y%yyyy%

The **Y%yyyy%** of the *nt:unstructured* type has the name beginning with the Y character followed by the year name having 4 numbers. It contains all the child nodes of **M%mm%**.

The **M%mm%** of the *nt:unstructured* type has the name beginning with the M character followed by the month name having 2 numbers. It contains all the child nodes of **D%dd%**.

The **D%dd%** of the *nt:unstructured* type has the name beginning with the D character followed by the date having 2 numbers. This node has two child nodes: **reminder** and **events**.

The **reminder** node of the *nt:unstructured* type contains the child nodes named basing on the Id of the event. This child node also has the nt:unstructured type. Each node is used to classify reminders of the same event. Each reminder is stored under a node of the *exo:reminder* type: *CalendarApplication/Y%yyyy%/M%mm%/D%dd%/reminders/%event_id%/%reminder_id%*.

The *exo:reminder* node type has the following properties:

Property name	Required type	Multiple	Description
exo:id	String	false	The Id of the reminder.
exo:eventId	String	false	The event Id of the reminder.
exo:creator	String	false	Define who creates the reminder.
exo:alarmBefore	Long	false	The amount of time that the reminder message is sent before the event starts.
exo:email	String	false	The list of emails to which the reminder

Property name	Required type	Multiple	Description
			message is sent.
exo:timeInterval	Long	false	Interval for resending the reminder message in minutes.
exo:reminderType	String	false	The types of reminders, including email and pop-up.
exo:fromDateTime	Date	false	The start time to send the reminder.
exo:remindDateTime	Date	false	The time to send the reminder.
exo:isRepeat	Boolean	false	Check if the reminder is repeated or not.
exo:isOver	Boolean	false	Check if the reminder is expired or not.
exo:summary	String	false	The summary of the reminder.
exo:description	String	false	The brief description of the reminder.

The **events** node of the *nt:unstructured* type contains the child node of the *exo:calendarPublicEvent* type defined at the path: *CalendarApplication/Y%yyyy%/M%mm%/D%dd%/events/%event_id%*.

Property name	Required type	Multiple	Description
exo:id	String	false	The Id of the public event.
exo:eventType	String	false	Event type, including Task and Event.
exo:calendarId	String	false	The calendar Id of the public event.
exo:rootEventId	String	false	The Id of each corresponding node: exo:calendarEvent.
exo:fromDateTime	Date	false	The start time of the public event.
exo:toDateTime	Date	false	The snd time of the public event.
exo:participant	String	true	The list of users being invited to the public event.

Property name	Required type	Multiple	Description
exo:eventState	String	false	The state of the public event, including: busy, available, outside.

The **events** node can add the **exo:repeatCalendarEvent** mixin node that has the following properties:

Property name	Required type	Multiple	Description
exo:repeatCount	Long	false	The number of times that the event is repeated.
exo:repeatUntil	Date	false	The given time until when the event is repeated.
exo:repeatInterval	Long	false	The interval when the event is repeated. It can be day, week, month or year corresponding to the repetition type chosen of day, week, month or year.
exo:repeatByDay	String	true	The given days in a week on which the event is repeated.
exo:repeatByMonthDay	Long	true	The given day/date in a month on which the event is repeated.
exo:recurrenceId	String	false	The Id of each event in the event series.
exo:excludeId	String	true	The Id of the events that are removed from the event series.
exo:isException	Boolean	false	Show whether the event is the exception in the event series or not. This case occurs when the event is removed from the repeated event series.
exo:originalReference	Reference	false	The UUID of the event that is repeated first.
exo:repeatFinishDate	Date	false	The end date on which the event is repeated.

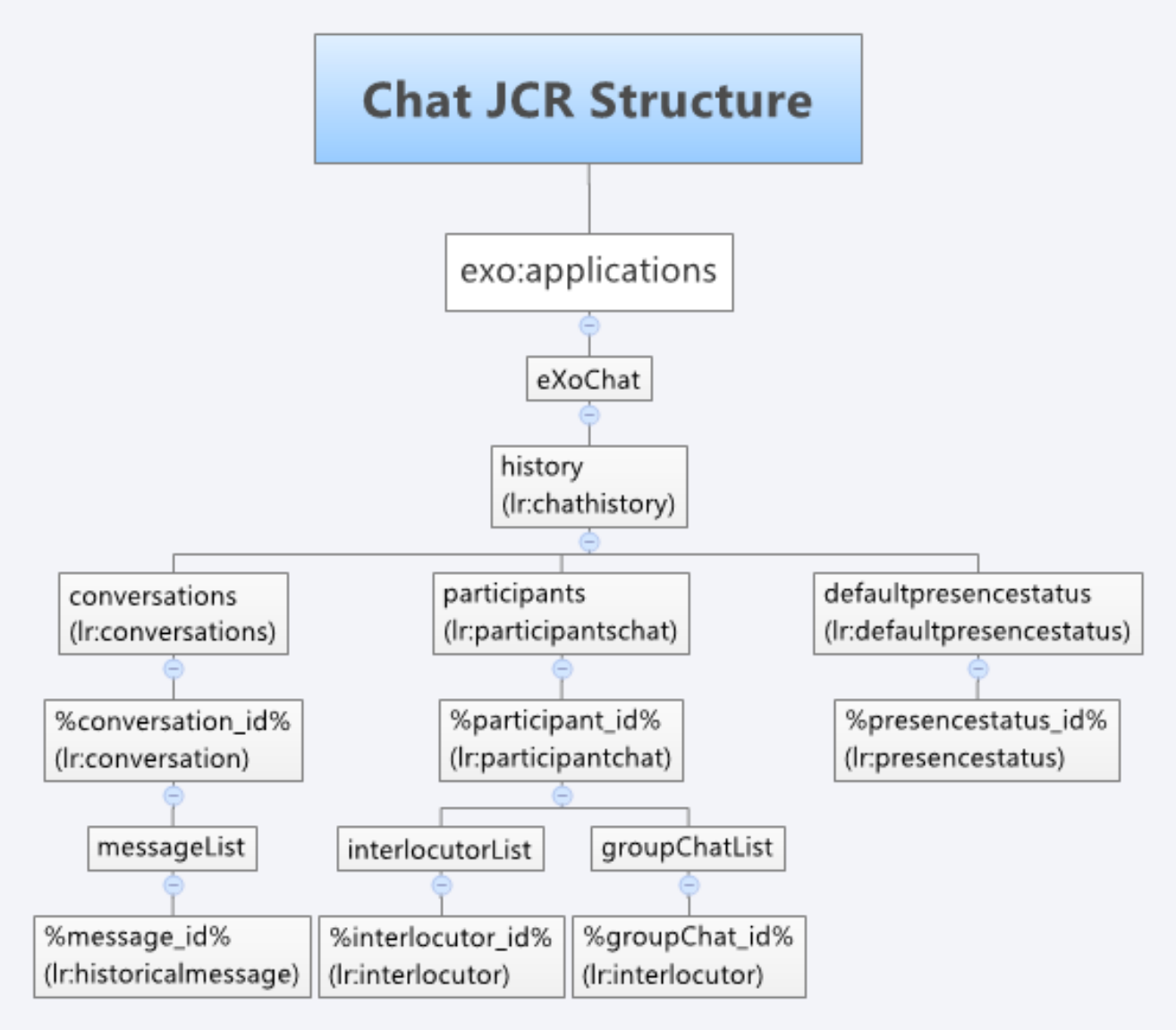
4.1.6. calendarSetting

The **calendarSetting** node of the *exo:calendarSetting* type is stored in *CalendarApplication/calendarsetting*. The *exo:calendarSetting* node type has the following properties:

Property name	Required type	Multiple	Description
exo:viewType	String	false	View type of the calendar. For more details, refer to the *org.exoplatform.calendar.service.CalendarSetting class.*
exo:timeInterval	Long	false	The interval for each action displayed each UI, for example, dragging and dropping one event in the Calendar application.
exo:weekStartOn	String	false	Define the start date of one week, complying with the *org.exoplatform.calendar.service.CalendarSetting class.*
exo:dateFormat	String	false	Define the date format, including dd/MM/yyyy, dd-MM-yyyy, MM/dd/yyyy, and MM-dd-yyyy.
exo:timeFormat	String	false	Define the time format, including "hh:mm a" and "HH:mm".
exo:location	String	false	Location where the calendar is set in format of the uppercase ISO 3166 3-letter country code.
exo:timeZone	String	false	The Id of the time zone, which is set by the user in compliance with the Java class: java.util.TimeZone.
exo:showWorkingTime	false	Boolean	Check if the working period is displayed or not.
exo:workingTimeBegin	String	false	Time to start working. This property only takes effect when exo:showWorkingTime is set to true.
exo:workingTimeEnd	String	false	Time to end working. This property only takes

Property name	Required type	Multiple	Description
			effect when exo:showWorkingTime is set to true.
exo:defaultPrivateCalendars	String	true	The list of the hidden private calendars.
exo:defaultPublicCalendars	String	true	The list of the hidden public calendars.
exo:defaultSharedCalendars	String	true	The list of the hidden shared calendars.
exo:sharedCalendarsColors	String	true	Define the color of the shared calendar, which is in the format of calendar id:color name.
exo:sendOption	String	false	The option to notify users before sending an invitation via email: never (not sending all time), always (sending message without asking) and ask (asking before sending).

4.2. Chat JCR Structure



The node type **lr:conversation** has the following properties:

Property name	Required type	Description
lr:conversationstartDate	Date	Start date of the conversation.
lr:conversationlastActiveDate	Date	Last date when the conversation is updated.

The node type **lr:historicalmessage** has the following properties:

Property name	Required type	Description
lr:messagefrom	String	Jabber Id of the user (or chat room) sending (or containing) the message respectively.
lr:messageto	String	Jabber Id of the user (or chat room) to whom (to which) the message is sent.
lr:messagetype	String	List of message types. For more details, refer to the

Property name	Required type	Description
		org.jivesoftware.smack.packet.Message.Type class.
lr:messagebody	String	Main content of the message.
lr:messagedateSend	Date	Date when the message was sent.
lr:messagereceive	Boolean	Check if the message has been received or not.

The node type **lr:participantchat** has the following properties:

Property name	Required type	Description
lr:participantchatjid	String	Jabber Id of the user.
lr:participantchatusername	String	Username of the portal.

The node type **lr:interlocutor** contains information regarding to the conversation between two users or of the chat room. It has the following properties:

Property name	Required type	Description
lr:conversationId	String	Id of the conversation which is the JCR node name of lr:conversation.
lr:interlocutorjid	String	Jabber Id of the chat room or user.
lr:interlocutorname	String	Username or name of the chat room.
lr:interlocutorisRoom	Boolean	Define if the conversation is performed between two users or is of chat room.

The node type **lr:defaultpresencestatus** has the following properties:

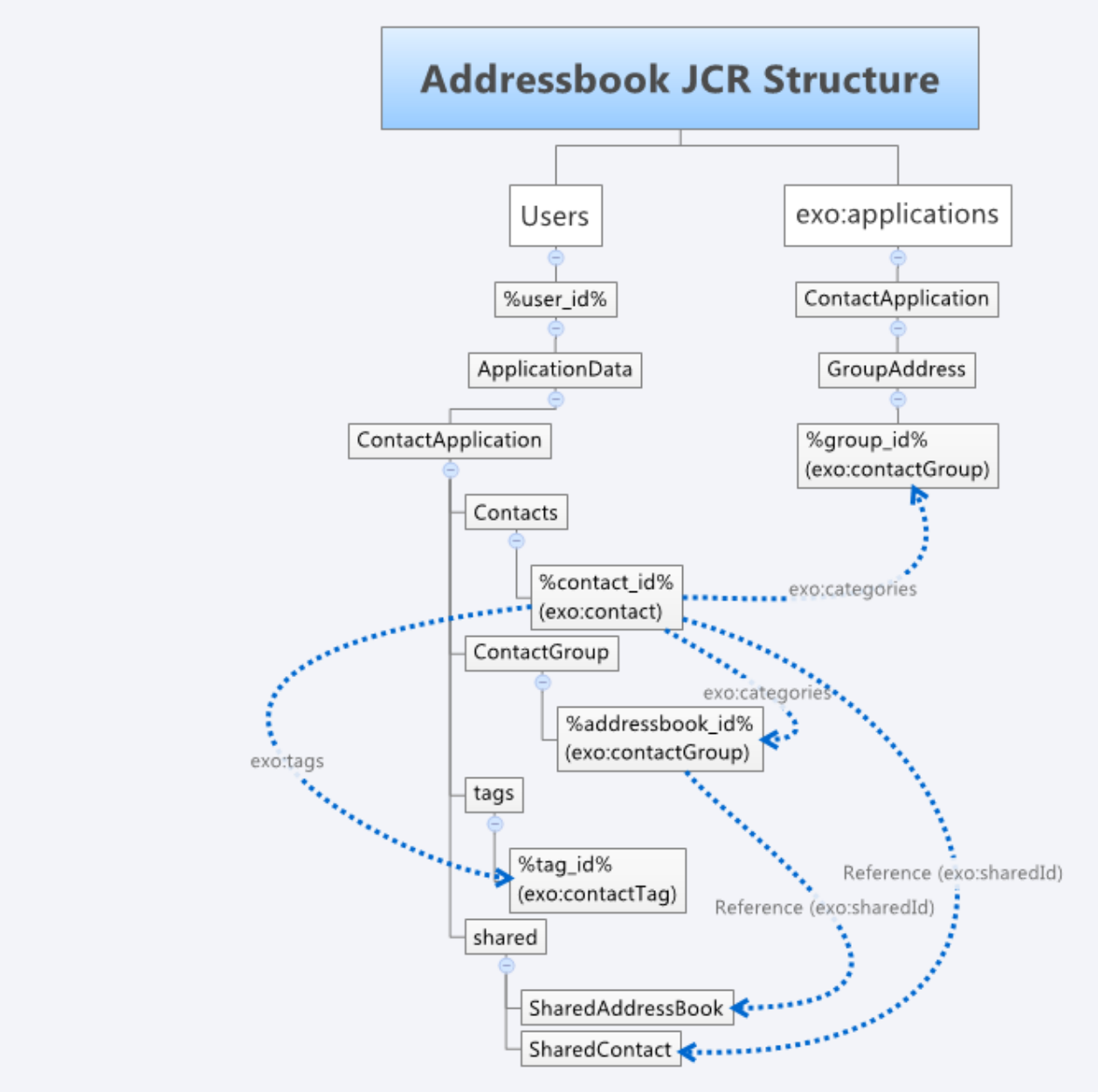
Property name	Required type	Description
lr:conversationlastActiveDate	Date	Date when the conversation is last updated.

The node type **lr:presencestatus** contains information regarding to the current status of user. It has the following properties:

Property name	Required type	Description
lr:userid	String	Id of the user.
lr:status	String	Current status of the user included in the *org.jivesoftware.smack.packet.Presence.Type class.*

4.3. Address Book JCR Structure

The Address Book portlet is a JCR-based application. The Address Book data are stored in the eXo-JCR under the ContactApplication data directory. The whole JCR structure of Address Book can be visualized in the diagram below:



4.3.1. Contacts

The **Contacts** node of the *nt:unstructured* type has the child nodes of the *exo:contact*. When a contact is created and added to an address book, it is stored in a node defined at the path: *ContactAppication/Contacts/%contact_id%*. When a contact is tagged, it is saved in the **tags** node with the *exo:tags* property. In case, a contact is shared to other users, it is referred to the **SharedContact** node of the *exo:sharedId* type. The *exo:contact* node type has the following properties:

Property name	Required type	Multiple	Description
exo:id	String	false	Node name of the

Property name	Required type	Multiple	Description
			exo:contact property.
exo:fullName	String	false	The full name of the contact.
exo:firstName	String	false	The first name of the contact.
exo:lastName	String	false	The last name of the contact.
exo:nickName	String	false	The nickname of the contact.
exo:gender	String	false	The gender of the contact.
exo:birthday	Date	false	The birthday of the contact.
exo:jobTitle	String	false	The job title of the contact.
exo:emailAddress	String	true	The email address of the contact.
exo:exoId	String	false	The Id of the user in the Chat application of eXo Collaboration.
exo:googleId	String	false	The Google Id of the user.
exo:msnId	String	false	The MSN Id of the user.
exo:aolId	String	false	The AOL Id of the user.
exo:yahooId	String	false	The Yahoo Id of the user.
exo:icrId	String	false	The ICR Id of the user.
exo:skypeId	String	false	The Skype Id of the user.
exo:icqId	String	false	The ICQ Id of the user.
exo:homeAddress	String	false	The home address of the contact.
exo:homeCity	String	false	The home city of the contact.
exo:homeState_province	String	false	The home state/province of the contact.
exo:homePostalCode	String	false	The home postal code of the contact.
exo:homeCountry	String	false	The home country of the contact.
exo:homePhone1	String	false	The primary home phone

Property name	Required type	Multiple	Description
			number of the contact.
exo:homePhone2	String	false	The secondary home phone number of the contact.
exo:homeFax	String	false	The home fax of the contact.
exo:personalSite	String	false	The personal site of the contact.
exo:workAddress	String	false	The address where the contact works.
exo:workCity	String	false	The city where the contact works.
exo:workState_province	String	false	The state/province where the contact works.
exo:workPostalCode	String	false	The postal code of the location where the contact works.
exo:workCountry	String	false	The country where the contact works.
exo:workPhone1	String	false	The primary phone number at the contact's working location.
exo:workPhone2	String	false	The secondary phone number at the contact's working location.
exo:workFax	String	false	The fax number at the contact's working location.
exo:mobilePhone	String	false	The mobile phone number of the contact.
exo:webPage	String	false	The website of the contact.
exo:note	String	false	The note about the contact.
exo:categories	String	true	The list of categories created by the user.
exo:editPermissionUsers	String	true	The list of users obtaining the edit permission.
exo:viewPermissionUsers	String	true	The list of users obtaining the view permission.

Property name	Required type	Multiple	Description
exo:editPermissionGroups	String	true	The list of groups obtaining the edit permission.
exo:viewPermissionGroups	String	true	The list of groups obtaining the view permission.
exo:tags	String	true	The list of tag Ids which the contact has marked.
exo:lastUpdated	Date	false	The time when the contact is last updated.
exo:isOwner	Boolean	false	Show if this contact is the contact of the user in the system or not.
exo:ownerId	String	false	If the contact is a user in the system, the owner Id will be the username of this user.

This node type may add the *exo:contactShared* mixin that has the following properties:

Property name	Required type	Multiple	Description
exo:sharedUserId	String	false	The name of the user sharing the contact.
exo:sharedId	Reference	true	The list of the references to shared users/groups.s

4.3.2. ContactGroup

The **ContactGroup** node of the *nt:unstructured* type contains all the child nodes of the *exo:contactGroup* type. These child nodes store address books and are stored in *ContactApplication/ContactGroup/%addressbook_id*. This node is also referred to the *exo:contact* node type with the *exo:categories* property. When an address book is shared, it is referred to the **SharedAddressBook** node with the *exo:sharedId* property.

The information of each address book is contained in a node of the *exo:contactGroup* type that has the following properties:

Property name	Required type	Multiple	Description
exo:id	String	false	The Id of the address book.
exo:name	String	false	The name of the address book.
exo:description	String	false	The brief description of

Property name	Required type	Multiple	Description
			the address book.
exo:editPermissionUsers	String	true	The list of users having the permission to edit the address book.
exo:viewPermissionUsers	String	true	The list of users having the permission to view the address book.
exo:editPermissionGroups	String	true	The list of groups having the permission to edit the address book.
exo:viewPermissionGroups	String	true	The list of groups having the permission to view the address book.

4.3.3. tags

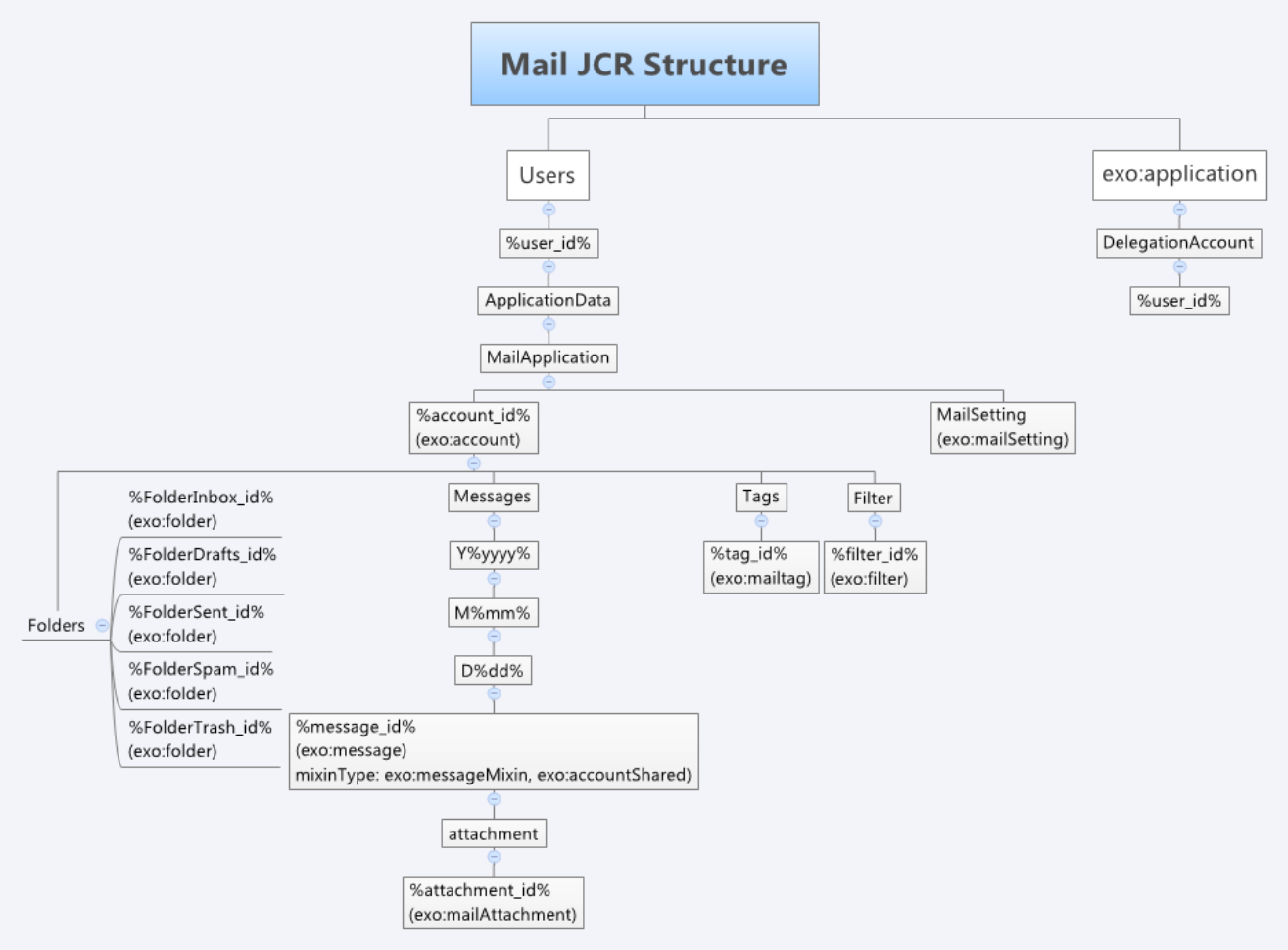
Tags are used to categorize the contacts in groups, so users can easily find the contacts. The **tags** node of the *nt:unstructured* type contains the child nodes of the *exo:contactTag* type. When a new tag is created, it is stored in a node defined at the path: *ApplicationData/ContactApplications/tags/%tags_id%*. The information of each tag is contained in a node of the *exo:contactTag* type that has the following properties:

Property name	Required type	Multiple	Description
exo:id	String	false	The Id of the tag.
exo:name	String	false	The name of the tag.
exo:description	String	false	The brief description of the tag.
exo:color	String	false	The color of the tag.

4.3.4. Shared

When a contact or a address book is shared to other users, it is referred to the **SharedContact** or **SharedAddressBook** node. Both of these child nodes have the *nt:unstructured* type.

4.4. Mail JCR Structure



The node type **exo:account** has the following properties:

Property name	Required type	Description
exo:id	String	Id of the account.
exo:label	String	Name of the account.
exo:userDisplayName	String	Screen name of the user.
exo:emailAddress	String	Email address of the account.
exo:emailReplyAddress	String	Email address of the account receiving replies.
exo:signature	String	Signature of the account.
exo:description	String	Brief description of the account.
exo:checkMailAuto	Boolean	Define if the mail is automatically checked after a given period or not.
exo:emptyTrash	Boolean	Define if the trash needs to be cleaned up when exiting from the Mail application or not.
exo:serverProperties	String	Information of the POP/IMAP server configuration.
exo:smtpServerProperties	String	Information of the SMTP server configuration.

Property name	Required type	Description
exo:lastCheckedTime	Date	Time when the account was last checked.
exo:checkAll	Boolean	Define if all folders of the mail are checked or not.
exo:checkFromDate	Date	Get mails as from the given date only if the value of exo:serverProperties is set for configuring the IMAP server.
exo:isSavePassword	Boolean	Define if the password is saved or not.
exo:secureAuthsIncoming	String	Type of the incoming connection for security. Its values include starttls, ssl/tls.
exo:secureAuthsOutgoing	String	Type of the outgoing connection for security. Its values include starttls, ssl/tls.
exo:authMechsIncoming	String	Authentication mechanism of the incoming connections. Its values consist of ntlm, plain, login, digest-md5, kerberos/gssapi, cram-md5.
exo:authMechsOutgoing	String	Authentication mechanism of the outgoing connections. Its values consist of ntlm, plain, login, digest-md5, kerberos/gssapi, cram-md5.
exo:permissions	String	Permissions of delegators.

The node type **exo:folder** has the following properties:

Property name	Required type	Description
exo:id	String	Id of the folder.
exo:name	String	Name of the folder.
exo:label	String	Absolute path referring to the folder on the Mail server.
exo:unreadMessages	Long	Number of unread messages in the folder.
exo:totalMessages	Long	Total number of messages in the folder.
exo:personal	Boolean	Define if the folder is created by one user or the Mail system.

Property name	Required type	Description
exo:folderType	Long	Type of folder, which is defined in the javax.mail.Folder class.
exo:lastStartCheckingTime	Date	Start time of the last check in the folder.
exo:lastCheckedTime	Date	End time of the last check in the folder.

The node type **exo:message** has the following properties:

Property name	Required type	Description
exo:id	String	Id of the message.
exo:uid	String	Id of the message on the IMAP server.
exo:inReplyToHeader	String	Id of the first message in the matching thread.
exo:path	String	Absolute path of the exo:message type.
exo:account	String	Id of the account.
exo:from	String	Value given in the **From** field in the email message, containing information of the sender, such as full name and email.
exo:to	String	Value given in the **To** field in the email message, containing information of the receiver, such as full name and email.
exo:cc	String	Value given in the **CC** field in the email message, containing information of the receivers, such as full name and email.
exo:replyto	String	Value given in the **Reply-To** field in the email message, such as emails.
exo:isUnread	Boolean	Define if the email has been read or not.
exo:subject	String	Subject of the email message that can be read from the **Subject** field.
exo:body	String	Main content of the email message.
exo:sendDate	Date	Date when the email message was sent.

Property name	Required type	Description
exo:receivedDate	Date	Date when the email message was received.
exo:size	Long	Capacity of the email message in bytes.
exo:contentType	String	Content type of the email message, for example: text/plain and text/html.
exo:folders	String	List of folder Ids containing the email message.
exo:tags	String	List of tag Ids marked in the email message.
exo:star	Boolean	Define if the email message is starred or not.
exo:hasAttach	Boolean	Define if any files are attached with the email message or not.
exo:priority	Long	Preference order of the message with 3 default values: 1 = High, 3 = Normal, 5 = Low.
exo:lastUpdateTime	Date	Time when the message was last updated.

The node type **exo:mailAttachment** has the following property:

Property name	Required type	Description
exo:fileName	String	Name of the file attached in the mail.

The node type **exo:mailtag** has the following properties:

Property name	Required type	Description
exo:id	String	Tag id of the mail.
exo:name	String	Name of the tag.
exo:description	String	Brief description of the mail tag.
exo:color	String	Color of the tag which is defined in the *org.exoplatform.webui.form.ext.UIFormColorP* class.

The node type **exo:filter** has the following properties:

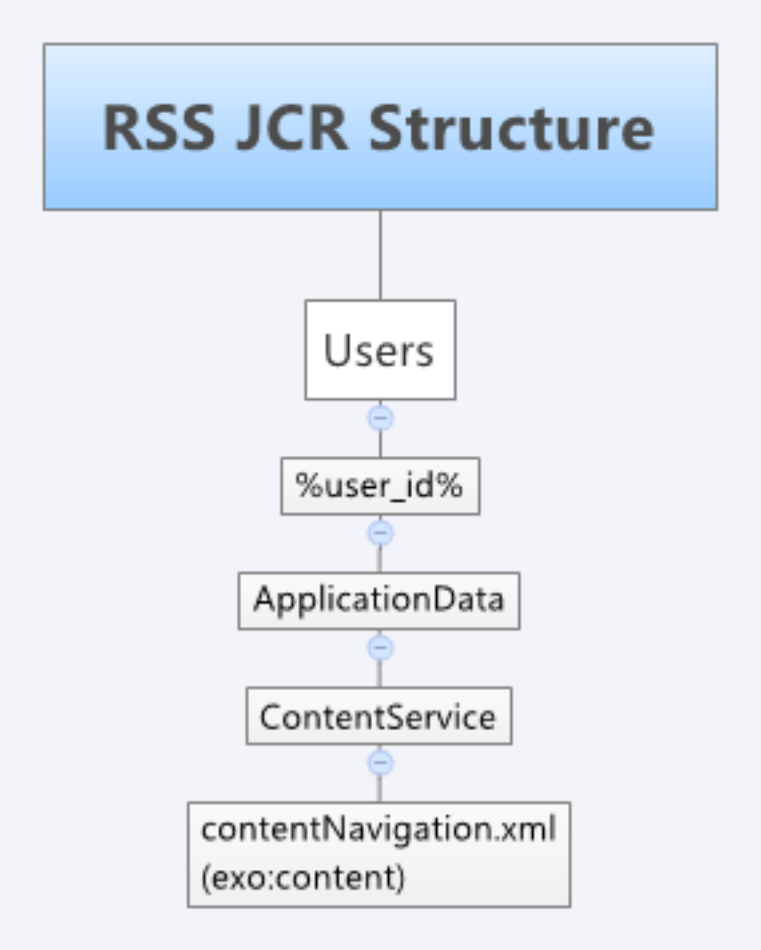
Property name	Required type	Description
exo:id	String	Filter id which is a unique and randomized value.
exo:name	String	Name of the filter which is defined by the user.
exo:from exo:to exo:subject exo:body	String	Filter email messages by each field respectively: * From * To * Subject * Body
exo:fromCondition exo:toCondition exo:subjectCondition exo:bodyCondition	Long	Filter emails by the condition types set in each property respectively: * exo:from * exo:to * exo:subject * exo:body All these properties have two values: * 0 = returned messages contains the value set in the corresponding property. * 1 = do not contain the value set in the corresponding property.
exo:applyTag	String	Apply the tag for the filtered email messages.
exo:applyFolder	String	Apply the folder for the filtered email messages.
exo:keepInbox	Boolean	Define if the email message is still

Property name	Required type	Description
		kept in the Inbox folder or not.
exo:applyForAll	Boolean	If the value is set to "true" into the exo:applyForAll property, the filter will be executed for all email messages.

The node type **exo:mailSetting** has the following properties:

Property name	Required type	Description
exo:numberMsgPerPage	Long	Number of messages displayed in one page.
exo:formatAsOriginal	Boolean	Define if the email message got from the mail server is kept in the original format or not.
exo:replyWithAttach	Boolean	Make the original message as the attachment before replying or not.
exo:forwardWithAttach	Boolean	Make the original message as the attachment before forwarding or not.
exo:prefixMsgWith	String	Prefix for the message.
exo:periodCheckAuto	Long	Time interval to check the email messages automatically.
exo:defaultAccount	String	Id of the user account that is displayed by default when the user logged in the Mail application.
exo:useWysiwyg	String	Define the Wysiwyg editor is used or not.
exo:saveMsgInSent	Boolean	Define the sent email message is saved to the Sent folder or not.
exo:layout	Long	Type of layout which is displayed to the user.
exo:returnReceipt	Long	Action type of the user when receiving the "return receipt" to confirm the arrival of one email message, including: 0 = ask, 1 = never, 3 = always.

4.5. RSS JCR Structure



The node type **exo:content** has the following properties:

Property name	**Required type**	**Description**
id	String	Id of the content.
ownerType	String	Type of the owner. Its default value is user.
ownerId	String	User Id of owner.
dataType	String	Type of data.
data	String	XML string of the content navigation.
createdDate	Date	Created date of the content.
modifiedDate	Date	Modified date of the content.

Chapter 5. Developer reference

5.1. Extension points

There are some overridable components in eXo Collaboration so that you can control how these components work by implementing or extending default implementations and then reconfigure these new components in the *configuration.xml* file.

5.1.1. ContentDAO

ContentDAO is an overridable component used in the Content application (or called RSS reader) of eXo Collaboration.

You can find the configuration file at *WEB-INF/cs-extension/cs/content/content-service-configuration.xml* with the declaration as below:

```
<component>
  <key>org.exoplatform.content.service.ContentDAO</key>
  <type>org.exoplatform.content.service.impl.ContentDAOImpl</type>
</component>
```

The example below is an example of plugin configuration:

```
<external-component-plugins>
  <target-component>org.exoplatform.content.service.ContentDAO</target-component>
  <component-plugin>
    <name>rssreader.listener</name>
    <set-method>addPlugin</set-method>
    <type>org.exoplatform.content.service.RSSContentPlugin</type>
    <description>rss reader plugin</description>
  </component-plugin>

  <component-plugin>
    <name>description.listener</name>
    <set-method>addPlugin</set-method>
    <type>org.exoplatform.content.service.DescriptionPlugin</type>
    <description>Description plugin</description>
  </component-plugin>
</external-component-plugins>
```

5.1.2. ContactLifeCycle

ContactLifeCycle is an interface to extend the capabilities of eXo Platform. The *ContactLifeCycle* lets you be notified during the lifecycle of an address book's contact when a contact is added or modified.

An example of *ContactLifeCycle* has been implemented to integrate the Address Book application in the eXo Social's Spaces. See the following configuration at: *ext/social-integration/src/main/resources/conf/portal/configuration.xml.*

See the following example:

```
<external-component-plugins>
  <target-component>org.exoplatform.contact.service.ContactService</target-component>
  <component-plugin>
    <name>ContactEventListener</name>
```

```
    <set-method>addListenerPlugin</set-method>
    <type>org.exoplatform.cs.ext.impl.ContactSpaceActivityPublisher</type>
  </component-plugin>
</external-component-plugins>
```

Details: `ContactSpaceActivityPublisher` implements `ContactLifeCycle`. This implementation publishes activities in the space activity stream to notify of new and updated contacts in the space address book.

5.1.3. Transport

Transport is an overridable component used in the Chat application of eXo Collaboration.

This overridable component is used to help users add the protocol to the Chat application, such as: ICQ, YAHOO, MSN, XMPP, AIM, GTALk.

The Chat application of eXo Collaboration only uses the XMPP protocol that is implemented in the *XMPPTransport* object.

5.1.4. EventLifeCycle

EventLifeCycle is an extension point used in the Calendar application of eXo Collaboration. You can find the configuration file of this component at: *ext/social-integration/src/main/resources/conf/portal/configuration.xml*.

See the following example:

```
<external-component-plugins>
  <target-component>org.exoplatform.calendar.service.CalendarService</target-component>
  <component-plugin>
    <name>CalendarEventListener</name>
    <set-method>addEventListenerPlugin</set-method>
    <type>org.exoplatform.cs.ext.impl.CalendarSpaceActivityPublisher</type>
  </component-plugin>
</external-component-plugins>
```

Details: `CalendarSpaceActivityPublisher` implements `EventLifeCycle`. It writes activities in the space activity stream when events or tasks are added/modified.

5.2. Public REST APIs

eXo Collaboration implements and provides many public REST APIS to help built-in applications, such as Calendar, Chat and Mail to communicate and transfer data with the server. By using these public REST APIs, extended or 3rd party applications can make use of this to develop cool web applications faster without much implementation.

5.2.1. Calendar application

The Calendar application of eXo Collaboration uses CalendarWebservice to provide all APIs for working with calendars, such as creating personal/group calendars, sharing calendars, managing events/tasks.

The REST API of Calendar portlet is exposed by *org.exoplatform.services.cs.calendar.CalendarWebservice* class.

Service name	Service URL	Location	Description
CalendarWebservice	`$portalname/$restcontextname/private/cs/calendar`	- Maven groupId: org.exoplatform.cs - ArtifactId: exo.cs.web.webservice	Call extended services of the Calendar application.

Details:

- $portalname: The name of the portal
- $restcontextname: The context name of rest webapplication which is deployed to the "$portalname" portal.
- Private: is optional and used for the protected access only. Such calls will require authentication.

- **APIs usage**:

Use the following APIs to build all functions of the Calendar application:

Name	Service URL endpoint	Parameters	Expected Values	Description
checkpermission	/private/cs/calendar/checkPermission/{username}/{calendarId}/type	username calendarid type	user id calendar id INVALID_TYPE = -1 / PRIVATE_TYPE = 0 / SHARED_TYPE = 1 / PUBLIC_TYPE = 2	Check the permission of a user to a calendar, aiming at defining if the user has the edit permission to the calendar or not.
event	/private/cs/calendar/event/{username}/{eventFeedName}	username eventFeedName	user id string	Return a feed RSS that lists links to access a specific event.
feed	/private/cs/calendar/feed/{username}/{feedname}/{filename}	username feedname filename	username string string	Show the content of a feed that is a list of events in the "filename" file.
publicProcess	/cs/calendar/subscribe/{username}/{calendarId}/{type}	username	user id	Process the public

Name	Service URL endpoint	Parameters	Expected Values	Description
		calendarId type	calendar id INVALID_TYPE = -1 / PRIVATE_TYPE = 0 / SHARED_TYPE = 1 / PUBLIC_TYPE = 2	calendar when having a remote access request.
privateProcess	/private/cs/calendar/p	rivate/{username}/{calendarId}/{type} username calendarID type	user id calendar id INVALID_TYPE = -1 / PRIVATE_TYPE = 0 / SHARED_TYPE = 1 / PUBLIC_TYPE = 2	Process the public calendar when having a remote access request. User must enter username and password to access.
upcomingEvent	/private/cs/calendar/g	etissues/{currentdatetime}/{type}/{limit} currentdatatime type limit	valid time format INVALID_TYPE = -1 / PRIVATE_TYPE = 0 / SHARED_TYPE = 1 / PUBLIC_TYPE = 2 integer	The list of upcoming events in a specific calendar.
updateStatus	/private/cs/calendar/u	pdatetaskstatus/{taskid}	task id	Update the status of a task.
getCalendars	/private/cs/calendar/g	etcalendars		Get a list of calendars.

5.2.2. Mail application

The Mail application of eXo Collaboration uses MailWebservice to provide all APIs for working with mail, such as sending/checking/storing mail to JCR.

REST API of the Mail portlet is exposed by *org.exoplatform.services.cs.mail.MailWebservice* class.

Service name	Service URL	Location	Description
MailWebservice	`$portalname/$restcontextname/private/cs/mail`	- Maven groupId: org.exoplatform.cs - artifactId: exo.cs.web.webservice	This service is used to call extended services of the Mail application.

- **APIs Usage**:

Use the following APIs to build all functions of the Mail application:

Name	Service URL endpoint	Parameters	Expected Values	Description
checkMail	/checkmail/{username}/{accountId}/{folderId/}/	username accountID folderId	user id account id folder id	Check mails when having a request.
synchFolders	/synchfolders/{username}/{accountId}/	username accountID	user id account id	Synchronize the mail folders in the clients and those in the mail server.
stopCheckMail	/stopcheckmail/{username}/{accountId}/	username accountId	user id account id	Stop checking the mail.
getCheckMailJobInfo	/checkmailjobinfo/{username}/{accountId}/	username accountId	user id account id	The method to get information of the mail-checking job.
searchemail	/searchemail/{keywords}	keywords	string	Search information from emails.

5.2.3. Chat application

The Chat application uses some APIs to help users create a room, join a room, invite other users to room, or send files, and more.

The Chat application of eXo Collaboration uses two services: RESTXMPPService and FileExchangeService to

do these tasks.

5.2.3.1. RESTXMPPService

REST API *RESTXMPPService* of the Chat portlet is exposed by *org.exoplatform.services.xmpp.rest.RESTXMPPService* class.

Service name	Service URL	Location	Description
RESTXMPPService	`$portalname/$restcontextname/xmpp`	* Maven groupid: org.exoplatform.cs * exo.cs.eXoApplication.chat.service	Implement all actions sent to the Chat server;

- **APIs Usage**:

Use the following APIs to build all functions of the Chat application:

Name	Service URL endpoint	Parameters	Expected Values	Description
loadJsResourceBundle	loadJsResourceBundle/{locale}/	locale	locale id	Read the language files in the Chat server.
createRoom	/muc/createroom/{username}/	username room nickname	user id room name display name	Create a chat room or a group chat.
configRoom	/muc/configroom/{username}/	username room	user id room name	Establish the configuration of a chat room.
getRoomConfigForm	/muc/getroomconfig/{username}/	username room	user id room name	Get the configuration of a chat room created.
getRoomInfo	/muc/getroominfo/{username}/	username room	user id string	Get the inforamtion of a chat room created.
getJoinedRooms	/muc/joinedrooms/{username}/	username	user id	List chat rooms that a user has been joined.

Name	Service URL endpoint	Parameters	Expected Values	Description
getRooms	/muc/rooms/{username}/	username	user id	Get a list of group chat or chat rooms created.
declineToRoom	/muc/decline/{username}/{inviter}/	username inviter room reason	user id user id room name string	Refuse the invitation to join the chat room.
destroyRoom	/muc/destroy/{username}	username room reason altroom	user id room name string room id	Delete a chat room created.
inviteToRoom	/muc/invite/{username}/{invitee}/	username invitee room reason	user id user id room name string	Invite other users to join a chat room.
joinRoom	/muc/join/{username}/	username room nickname password	user id room name display name room password	Join a chat room.
leftRoom	/muc/leaveroom/{username}/	username room	user id room name	Leave a chat room.
changeNickname	/muc/changenickname/{username}/{nickname}/			Change the

Name	Service URL endpoint	Parameters	Expected Values	Description
		username nickname	user id display name	nickname of users.
changeAvailabilityStatusInRoom	muc/changestatus/{username}/{mode}/	username mood room status	user id presence type room name presence type	Change the status of a user in the chat room.
changeSubject	muc/changesubject/{username}/	username room subject	user id room name string	Change the subject of a chat room.
manageRoleRoom	/muc/managerole/{username}/	username room nickname role command	user id room name display name Participant / moderator grant/revoke	Change the role of each user in a chat room.
manageAffilationRoom	/muc/manageaffiliation/{username}/	username room nickname affiliation command	user id room name display name String affiliation grant / revoke	Change the ownership of a chat room.
kickUserFromRoom	/muc/kick/{username}/			Remove a user from

Name	Service URL endpoint	Parameters	Expected Values	Description
		username nickname room reason	user id display name room name string	the chat room.
banUserFromRoom	/muc/ban/{username}/	username room name reason	user id room name user id string	Ban a user in the chat room.
addBoddyToRoster	/roster/add/{username}/{adduser}	username adduser nickname group	user id use id display name group id	Add a user into the contact list.
updateBoddy	/roster/update/{username}/{upduser}/	username upduser nickname group	user id user id display name group id	Update new users into the contact list.
createGroup	/roster/group/{username}/{group}/	username group	user id group id	Create a chat room.
askForSubscription	/askforsubscription/{username}/{askuser}/	username askuser	user id user id	Change the presence type of a user into Subscription.

Name	Service URL endpoint	Parameters	Expected Values	Description
		nickname	display name	
cleanBuddylist	/rosterclean/{username}/	username	user id	Remove a user from the contact list.
getAllHistory	/history/getmessages/{usernameto}/{isGroupChat}/	usernameto isGroupChat usernamefrom	user id true / false user id	Get all the chat history of two users.
getHistoryBetweenDate	/history/getmessages/{usernameto}/{isGroupChat}/{from}/{to}/	usernameto isgroupchat from to usernamefrom	user id true / false valid date format valid date format user id	Get the chat history of two users in a specific period.
getHistoryFromDateToNow	/history/getmessages/{usernameto}/{isGroupChat}/{from}/	username isGroupChat from usernamefrom	user id true / false valid date format valid date format user id	Get the chat history of two users from a specific date to the current time.
getAllHistoryFile	/history/file/getmessages/{usernameto}/{isGroupChat}/{clientTimezoneOffset}/	uernameto isGroupChat clientTimezoneOffset usernamefrom	user id true / false Long user id	Download all the chat history file of two users.
getHistoryFromDateToNowFile	/history/file/getmessages/{usernameto}/{isGroupChat}/{from}/{clientTimezoneOffset}/	usernameto	user id	Download the chat history file of two users from a

Name	Service URL endpoint	Parameters	Expected Values	Description
		isGroupChat from clientTimezoneOffset usernamefrom	true / false valid date format Long user id	specific date to the current time
getHistoryBetweenDateFile	/history/file/getmessages/{usernameto}/{isGroupChat}/{from}/{to}/{clientTimezoneOffset}/	usernameto isGroupChat from to clientTimezoneOffset usernamefrom	user id true / false valid date format valid date format Long user id	Download the chat history file of two users in the specific date.
getUserInfo	/getuserinfo/{username}/{needinfo}/	username needinfo	user id string	Get the information of a user.
login2	/login2/{forcache}/	forcache		Allow a user to log in the chat server.
logout	/logout/{username}/{presencestatus}/	username presencestatus	user id presencestatus	Allow a user to log out the chat server.
messageReceive	/history/messagereceive/{username}/{messageid}/	username messageid	user id message id	Receive a message from other users.
removeBuddy	/roster/del/{username}/{removeboddy}/	username removeboddy	user id user id	Delete a contact from the contact list.
removeTransport	/removetransport/{username}/{transport}/	username	user id	Reset the presence type at the service that is being used.

Name	Service URL endpoint	Parameters	Expected Values	Description
		transport	transport servive (e.g: Yahoo, XMPP)	
searchUsers	/searchuser/{username}/	username search byUsername byName byEmail searchService	user id string true / false true /false true / false string	Search users in the chat server.
sendMessage	/sendmessage/{username}/	username messageBean	usesr id object	Send an message to other users.
sendMUCMessage	/muc/sendmessage/{username}/	username messageBean	user id object	Send a message to multile users or a group.
setUserStatus	/sendstatus/{username}/{status}/	username status	user id available/ unavailabe / do not disturb / away / extend away	Change the status of a user.
subscribeUser	/subscribeuser/{username}/{subsuser}/	username subsuser	user id user id	Change presence type into Subcribed type.
unsubscribeUser	/unsubscribeuser/{username}/{unsubsuser}/	username unsubsuser	user id user id	Change presence type into the Unsubscribed type.
acceptFile	/fileexchange/accept/{username}/{uuid}/	username	user id	Accept getting a file sent from another user.

Name	Service URL endpoint	Parameters	Expected Values	Description
		uuid	string	
rejectFile	/fileexchange/reject/{username}/{uuid}/	username uuid	user id string	Refuse getting a file sent from another user.
getPreviousStatus	/getprevstatus/{username}/	username	user id	Get the tatus of a user in the last log-in.

5.2.3.2. FileExchangeService

REST API *FileExchangeService* for uploading files is defined in *org.exoplatform.services.xmpp.rest.FileExchangeService* class.

Service name	Service URL	Location	Description
FileExchangeService	`$portalname/$restcontextname/fileexchange`	- Maven groupid: org.exoplatform.cs - InterfactId: exo.cs.eXoApplication.chat.service	Upload a file to the server and inform the user that the file can be downloaded to the local computer.

- APIs usage: Use the following APIs to upload and send files to other users:

Name	Service URL endpoint	Parameters	Expected Values	Description
upload	$portalname/$restcontextname/fileexchange	description username requestor isroom	string user id user id true / false	Upload a file to the server.