

eXo Platform 3.0 - Collaboration Functions

Copyright ©

1. Prerequisites	1
2. Applications	2
2.1. Portlets	2
2.1.1. Calendar Portlet	2
2.1.2. Chatbar Portlet	2
2.1.3. Chat Portlet	5
2.1.4. Contact Portlet	5
2.1.5. Mail Portlet	5
2.1.6. RSSreader Portlet	6
2.2. Gadgets	6
2.2.1. Eventslist	6
2.2.2. Taskslst	7
2.2.3. Messageslist:	7
3. Configurations	9
3.1. Components in eXo Collaboration Configuration	9
3.1.1. CalendarService	9
3.1.2. HistoryImpl	10
3.1.3. XMPPMessenger	10
3.1.4. DefaultPresenceStatus	11
3.1.5. ContactService	12
3.2. External Component Plug-ins	12
3.2.1. Calendar Configuration	14
3.2.2. AddActionsPlugin	22
3.2.3. Chat Configuration	22
3.2.4. Contact Configuration	25
3.2.5. Content Configuration	27
3.2.6. Mail Configuration	28
3.2.7. Social Integration Configuration	28
3.3. eXo Chatserver Configuration	31
3.3.1. Openfire Configuration	31
3.3.2. System Configuration	34
3.3.3. AS configuration	34
4. Developer reference	36
4.1. Extension points	36
4.1.1. ContentDAO	36
4.1.2. ContactLifeCycle	36
4.1.3. Transport	37
4.1.4. EventLifeCycle	37
4.2. Public REST APIs	37
4.2.1. Calendar application	37
4.2.2. Mail application	40
4.2.3. Chat application	40

Chapter 1. Prerequisites

You can refer the following links to understand more about this developer reference guide:

- About GateIn:

<http://docs.jboss.com/gatein/portal/3.1.0-FINAL/reference-guide/en-US/html/index.html>

- About OpenSocial gadget:

http://wiki.opensocial.org/index.php?title=Main_Page

- About WebService:

<http://docs.jboss.org/exojcr/1.12.4-GA/developer/en-US/html/part-ws.html>

- About Chatserver:

<http://www.igniterealtime.org/builds/openfire/docs/latest/documentation/javadoc/index.html>

- About Vcard:

<http://en.wikipedia.org/wiki/VCard>

- About ICalendar standard:

<http://en.wikipedia.org/wiki/ICalendar> <http://www.ietf.org/rfc/rfc2445.txt>

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

Calendar Portlet shows the Calendar application of eXo Collaboration, that offers a lot of features to users.

The Calendar application includes the following features:

- Create multiple personal calendars, manage calendars easily with calendar groups.
- Quickly create an event or a task using the **Quick Add** dialog.
- Create events and 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 .
- Quickly view events day by day by navigating the mini-calendar.
- Share calendars with others.
- Import/Export calendars.
- Publish your calendars with RSS, CalDAV.
- Search events/tasks in calendars.
- Print your agenda.

2.1.1.2. Portlet.xml

To see the portlet in the project, please 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 browser window is scrolled or resized. It has a fixed height, but expands horizontally to whatever size is available in its container. This allows to place the portlet in two layout cases :

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

Chat Bar application implements all functions of Chat application with the purpose of allowing user to send and receive messages anywhere after he has logged in.

The Chat Bar is a typical tool-bar with buttons that open menus. It gives access to main chat features :

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

2.1.2.2. Portlet preferences

The Chatbar Portlet consists 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-->
    </preference>

</portlet-preferences>
```

```

        <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
        <read-only>true</read-only>
    </preference>
</portlet-preferences>

```

Preference Name	Possible Values	Default Values	Description
showMailLink	true / false	true	true or false value to let users see the application icon or not
showCalendarLink	true / false	true	true or false value to let user see the application icon or not
showContactLink	true / false	true	true or false value to let user see the application icon or not
mailUrl	string	Portal/private/intranet/page	page url to application page it come when you configuration page.xml and navigation.xml in portal no need to combine with %domain name : port% chatbar will resolve it from server
calendarUrl	string	Portal/private/intranet/page	page url to application page it come when you configuration page.xml and navigation.xml in portal no need to combine with %domain name : port% chatbar will resolve it from server
contactUrl	string	Portal/private/intranet/page	page url to application page it come when you configuration page.xml and navigation.xml in portal no need to combine with %domain name : port% chatbar will resolve it from server
info	Info	Info	This is only the key to get the resource bundle 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, eXo Mail 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, etc.

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.

2.2.1.2. Description of Preferences

See the 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="/calendar/subscribeurl" />
  <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 Calendar portlet
Subscribeurl	link to 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](#)

2.2.1.3. Links to used REST services

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

2.2.2. Tasklist

Tasklist list 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 of Preferences

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"
```

Details: Setting: It 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 of Preferences

See the preferences 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="/mail/subscribeurl" />
  <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 Mail Application
Subscribeurl	link to upcoming messages
Limit	The number of displayed unread messages that is set by users
Account	The mail account in Mail application
Folder	The folder consists of unread messages
Tag	The tags in all unread messages

2.2.3.2. Links to used REST services

It uses 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 the 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	Workspace is the workspace name in JCR where history data is stored	string	collaboration
repository	Repository is the repository name in JCR where history data is stored	string	repository
path	Path is 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	IP address or hostname for the openfire server	integer	127.0.0.1
	port	Port to connect to on the openfire server. Should be the same that set in openfire configuration "Client to Server"	integer	5222
send-file	timeout	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	It shows 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 that contains the declaration of ContactService component is executed, after that a ContactService component is created with params and plug-ins 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
      <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 shouldn't be displayed in broadcast list. Wildcards may be used in groups name	true/false	N/A

3.2. External Component Plug-ins

Applications	Components	Description
Calendar	NewUserListener	It is a plug-in used to create default personal calendars
	NewGroupListener	It is a plug-in used to create default group calendars
	NewMembershipListener	It is plug-in used to share calendars to members of a specific group

Applications	Components	Description
	ReminderPeriodJob	It is a plug-in used to execute sending reminder emails to users
	PopupReminderPeriodJob	It is a plug-in used to open a pop-up reminder on the browser of users
	AddActionsPlugin	It is a plug-in that helps the systems automatically update the updated date of events / tasks in a calendar when the content of these events / tasks are changed
Chat	HistoryPeriodJob	It is a plug-in used to save the chat history of users
	RequestFilterComponentPlugin	It is used to delete the session of a user when he suddenly closes the browser or changes the session
	AuthenticationLoginListener	It is used to start the session and log in the chat server
	AuthenticationLogoutListener	It is used to end the session and log out the chat server
Contact	NewUserListener	It is used to create personal contact data for users
	NewMembershipListener	It is used to create address book for a specific group
	UpdateUserProfileListener	It is used to update the personal profile of a user when he changes it on the portal
Content	RSSContentPlugin	It is a formatter used to analyze the data from a RSS rsource
	DescriptionPlugin	It is a plug-in to represent the data from a RSS source
Mail	AuthenticationLogoutListener	It is a plug-in used to stop checking mails of a user when he logs out
Social Intergration	CalendarDataInitialize	It is a plug-in used to create a calendar for a group in a specific space
	ContactDataInitialize	It is a plug-in used to create an address book for a group in a specific space
	ContactSpaceActivityPublisher	It is a plug-in used to customize the activity status of a specific space when an event happens on a

Applications	Components	Description
		address book
	CalendarSpaceActivityPublisher	It is a plug-in used to customize the activity status of a specific space when an event happens on a calendar
	PortletPreferenceRequiredPlugin	It is a plug-in used to 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 plug-in 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>
  </init-params>
</component-plugin>
```

```

<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>

```

Explanation:

- **Name:** `calendar.new.user.event.listener` - 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 plug-in used to create default personal calendars.

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

Value-params	Description	Possible value	Default Value
defaultEventCategories	Default event categories for users	String (Comma separated list of category names)	Meeting, Calls, Clients, Holiday, Ann
defaultCalendarCategory	Name of the calendar group	String	Default

Value-params	Description	Possible value	Default Value
viewType	Default view after user logins and goes to Calendar portlet	0-6 (see below)	1
timeInterval	The time unit interval when you drag and move the event (in Day view and Week view only)	integer in minutes	15
weekStartOn	Day to use as the beginning of the week. only affect on Week view	1-7 (see below)	2
dateFormat	The display format for dates	valid Java Date format	MM/dd/yyyy
timeFormat	The display format for time	valid Java Date format	HH:mm
localeId	ID of the geographic locale	valid locale ID	BEL
timezoneIds	User time zone	valid TimeZone id	Europe
baseUrlForRss	The url to publish the RSS content	none	none
isShowWorkingTime	Indicate if the working time should be highlighted in day view	true/false	false
workingTimeBegin	The start time in working time	time in timeFormat	08:00
workingTimeEnd	The end time in working time	time in timeFormat	18:00
ignoredUsers	Definition users to ignore create default calendar	user id, use multiple by each line	demo/marry

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

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 plug-in 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>
```

Explanation:

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

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

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

3.2.1.3. NewMembershipListener

To use the plug-in 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>
```

Explanation:

- **Name:** `calendar.new.membership.event.listener` is the unique key to avoid duplicate names. Users can change it.
- **Set-method:** `addListenerPlugin` is the function is executed at the target of the component.
- **Type:** `org.exoplatform.calendar.service.impl.NewMembershipListener` is the class is set up to execute the creation of database.
- **Description:** It is a plug-in used to execute sending reminder emails to users.

3.2.1.4. ReminderPeriodJob

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

You must use the following target component to use the plug-in 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>
```

Explanation:

- **Name:** `RecordsJob` is the name of a schedule job. Keep this value.
- **Set-method:** `addPeriodJob` is the plug-in registering method. Keep this value.
- **Type:** `org.exoplatform.calendar.service.ReminderPeriodJob` is a class that executes to transfer the data

into database of Job Scheduler.

- **Description:** Add email reminder job to the JobSchedulerService.

See the 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
		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 the job starts to run	Integer	+0
		endTime	The time the job ends to run	Integer	none

3.2.1.5. PopupReminderPeriodJob

You must use the following target component to use the plug-in 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>

```

Explanation:

- **Name:** PopupRecordsJob is the name of the job. Keep this value.
- **Set-method:** addPeriodJob is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.calendar.service.PopupReminderPeriodJob is a class that executes to transfer the data into database of Job Scheduler.
- **Description:** Add popup reminder job to the JobSchedulerService.

See the 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	PopupReminderJob	
		groupName	The name of group job	String	CollaborationSuite	
		job	The name of actual job class	Class path	org.exoplatform.calendar.service.PopupReminderPeriodJob	
		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 the job starts to run	Long	+0	
		endTime	The time the job ends to run	Integer	None	
popupreminder.info	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/web-service-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-param>
        </init-params>
      </component-plugin>
    </component-plugins>
  </component>
```

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

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

3.2.3. Chat Configuration

The Chat configuration 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.HistoryPeriodJob"/>
      <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>

```

Explanation:

- **Name:** ChatRecordsJob is the name of the job. Keep this value.
- **Set-method:** addPeriodJob is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.services.xmpp.connection.impl.HistoryPeriodJob is a class that 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 the 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	messageToHistoricalMessageJob
		groupName	The name of group name	String	CollaborationSuite
		job	The name of actual job class	Class path	org.exoplatform.services.xmpp.connection.impl.HistoryPeriodJob
		repeatCount	How many times to run this job	integer	0 (use '0' which means 'run forever')
		period	The time interval (millisecond) between job executions	Long	3000
		startTime	The time the job	Long	+0

Properties-param	Description	Property names	Description	Possible values	Default values
			starts running.		
		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>RESTMPPServiceFilter</name>
        <value>org.exoplatform.services.xmpp.rest.filter.RESTMPPServiceFilter</value>
      </value-param>
    </init-params>
  </component-plugin>
</external-component-plugins>

```

Explanation:

- **Name:** ws.rs.request.filter is the name of the filter. Keep this value.
- **Set-method:** addPlugin is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.services.rest.impl.RequestFilterComponentPlugin is a class that 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 the details about the init-params of the component in the following table:

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

3.2.3.3. AuthenticationLoginListener and AuthenticationLogoutListener

Two functions: login and logout of XMPPRestService is responsible for creating a new XMPPSessionImpl and destroying an existed XMPPSessionImpl. They can be called by listeners: AuthenticationLoginListener, AuthenticationLogoutListener or from client(browser) through Rest protocol (jabberLogin, jabberLogout in

UIMainChatWindow.js). You must use the same target component for two external component plug-ins:

```
<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>
```

Explanation:

- **Name:** `exo.core.security.ConversationRegistry.register` is the name of plug-in.
- **Set-method:** `addListener` is the plug-in registering method. Keep this value.
- **Type:** `org.exoplatform.services.xmpp.connection.impl.AuthenticationLoginListener` is a class to execute the requests of the plug-in.
- **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>
```

Explanation:

- **Name:** `exo.core.security.ConversationRegistry.unregister` is the name of plug-in.
- **Set-method:** `addListener` is the plug-in registering method. Keep this value.
- **Type:** `org.exoplatform.services.xmpp.connection.impl.AuthenticationLogoutListener` is a class to execute the requests of the plug-in.
- **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 plug-ins: `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>
```

Explanation:

- **Name:** contact.new.user.event.listener is the name of listener.
- **Set-method:** addListenerPlugin is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.contact.service.impl.NewUserListener is a class that executes all the requires of the plug-in.
- **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>
```

Explanation:

- **Name:** contact.new.membership.event.listener is the name of the listener.
- **Set-method:** addListenerPlugin is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.contact.service.impl.NewMembershipListener is a class that executes all the requires of the plug-in.
- **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>
```

Explanation:

- **Name:** `contact.new.userprofile.event.listener` is the name of the listener.
- **Set-method:** `addListenerPlugin` is the plug-in registering method. Keep this value.
- **Type:** `org.exoplatform.contact.service.impl.UpdateUserProfileListener` is a class that executes all the requires of the plug-in.
- **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 plug-ins: `RSSContentPlugin` and `DescriptionPlugin`. Both the external components plug-ins 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>
```

Explanation:

- **Name:** `rssreader.listener` is the name of the listener.
- **Set-method:** `addPlugin` is the plugin registering method. Keep this value.
- **Type:** `org.exoplatform.content.service.RSSContentPlugin` is a class extending `ContentPlugin` and implements `loadContentMeta` method to get content items.
- **Description:** It is a formatter 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>
```

Explanation:

- **Name:** `description.listener` is the name of the listener.
- **Set-method:** `addPlugin` is the plug-in registering method. Keep this value.
- **Type:** `org.exoplatform.content.service.DescriptionPlugin` is a class that executes all the requires of the plug-in.
- **Description:** It is a plug-in to represent the 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 check messages for one account, the remote mailbox fetch is performed as a background job. Before CS 1.2, the job was continuing 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 continuing. This can be resource intensive for the server if many users have large mailboxes.

Since CS 1.2, we added the capability to halt the background job when the user session terminates (logout or time out). It makes CS more friendly with server resources. If you want to activate this feature, you need to add a bunch of xml configuration 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>
```

Explanation:

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

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>

```

Explanation:

- **Name:** CalendarDataInitialize is the name of plug-in.
- **Set-method:** addSpaceListener is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.cs.ext.impl.CalendarDataInitialize is a class that executes all the requires of the plug-in.
- **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>

```

Explanation:

- **Name:** ContactDataInitialize is the name of the plug-in.
- **Set-method:** addSpaceListener is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.cs.ext.impl.ContactDataInitialize is a class that executes all the requires of the plug-in.
- **Description:** It is a plug-in 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	Description	Possible value	Default value
portletName	The name of the portlet	String	ContactPortlet

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>
```

Explanation:

- **Name:** ContactEventListener is the name of the plug-in.
- **Set-method:** addListenerPlugin is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.cs.ext.impl.ContactSpaceActivityPublisher is a class that executes all the requires of the plug-in.
- **Description:** It is a plug-in used to customize the activity status of a specific space when an event happens on a 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>
```

Explanation:

- **Name:** CalendarEventListener is the name of the plug-in.
- **Set-method:** addEventListenerPlugin is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.cs.ext.impl.CalendarSpaceActivityPublisher is a class that executes all the requires of the plug-in.
- **Description:** It is a plug-in 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>

```

Explanation:

- **Name:** ortlets.prefs.required is the name of the plug-in.
- **Set-method:** setPortletsPrefsRequired is the plug-in registering method. Keep this value.
- **Type:** org.exoplatform.social.core.application.PortletPreferenceRequiredPlugin is a class that executes all the requires of the plug-in.
- **Description:** It is a plug-in 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	Description	Possible value	Default value
portletsPrefsRequired	The name of plug-in added to SpaceService	String	ContactPortlet / ContactPortlet

3.3. eXo Chatserver Configuration

3.3.1. Openfire Configuration

eXo Collaboration chat services is a Jabber engine powered by Openfire . eXo will delegate the actual Jabber protocol communication to Openfire.

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

the admin console : <http://localhost:9090/> the openfire.xml file, in \$openfirehome/conf/

3.3.1.1. Configuration in Openfire.xml

The Openfire server has a single configuration file called openfire.xml and located under exo-openfire/conf directory. Configuration is based on properties expressed in an XML syntax. For example, to set 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>

3.3.1.2. eXo specific configuration

eXo Collaboration bundle comes with a pre-configured openfire server. It is bundled with some eXo plugins and configurations that allow connectivity with eXo. 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 provides implementations for these 3 interfaces with ExoAuthProvider, ExoUserProvider, ExoGroupProvider. These implementations are based on eXo REST framework and let you configure the endpoints within the openfire.xml file with additional properties :

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

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

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



Note

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

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

Example:

- If 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 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, Openfire server is dedicated to the portal named "portal"

3.3.2. 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 port used for respectively the unsecured and secured Openfire Admin Console access.
All addresses	7777	File Transfer Proxy	The port used for the proxy service that allows file transfers to occur between two entities on the XMPP network.
All addresses	3478 & 3479	STUN Service	The port used for the service that ensures connectivity between entities when behind a NAT.

You can view the table above in <http://hostname:9090/index.jsp> after you are logged in to openfire's web console and also customize those ports by yourself.

3.3.3. AS configuration

To enable propagation of identity across the chat webapp, it is required that you enable the SSO valve on Tomcat-based Application server.

- For jboss server, edit `jboss/server/default/deploy/jboss-web.deployer/server.xml`
- For 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 the case of cluster deployment, you may want to use `ClusteredSingleSignOn` instead.

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

Chapter 4. Developer reference

4.1. 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 file `configuration.xml`.

4.1.1. 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 the 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>
```

4.1.2. ContactLifeCycle

ContactLifeCycle is an interface that you can implement to extend the capabilities of eXo. A ContactLifeCycle lets you be notified during the lifecycle of an address book's contact :

- when a contact is added
- when a contact is modified

An example of ContactLifeCycle has been implemented to integrate Address Book application in 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>
```

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

4.1.3. 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, etc.

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

4.1.4. 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>
```

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

4.2. Public REST APIs

eXo Collaboration implements and provides many public REST APIs to help built-in applications such as Calender, 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.

4.2.1. Calendar application

Calendar application of eXo Collaboration use CalendarWebservice to provide all APIs for working with calendar such as create personal calendars, group calendars, share calendars, manage events / tasks, etc.

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

Service name	Service URL	Description	Location	
CalendarWebservice	<code>\$portalname/\$restcontextname/private/cs/calendar</code>	This service is used to call extended services of Calendar application.	- Maven groupId: <code>org.exoplatform.cs</code> - ArtifactId: <code>exo.cs.web.webservice</code>	



Note

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

• APIs usage:

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

Name	Description	Service endpoint	URL	Parameters	Expected Values
checkpermission	It checks the permission of a user to a calendar, aiming at defining if the user has the edit permission to the calendar or not.	<code>/private/cs/calendar/checkPermission/{username}/{calendarId}/type</code>		username calendarid type	user id calendar id INVALID_TYPE = -1 / PRIVATE_TYPE = 0 / SHARED_TYPE = 1 / PUBLIC_TYPE = 2
event	It returns a feed rss that lists links to access a specific event.	<code>/private/cs/calendar/event/{username}/{eventFeedName}</code>		username eventFeedName	user id string
feed	It shows the content of a feed that is a	<code>/private/cs/calendar/feed/{username}/{feedname}/{filename}</code>		username feedname filename	username filename

Name	Description	Service endpoint	URL	Parameters	Expected Values
	list of events in the "filename" file.			feedname filename	string string
publicProcess	It processes the public calendar when having a remote access request.	/cs/calendar/subscribe	{username}/{calendarId}/{type}	username calendarId type	user id calendar id INVALID_TYPE = -1 / PRIVATE_TYPE = 0 / SHARED_TYPE = 1 / PUBLIC_TYPE = 2
privateProcess	It processes the public calendar when having a remote access request. User must enter username and password to access.	/private/cs/calendar/private	{username}/{calendarId}/{type}	username calendarID type	user id calendar id INVALID_TYPE = -1 / PRIVATE_TYPE = 0 / SHARED_TYPE = 1 / PUBLIC_TYPE = 2
upcomingEvent	It is a list of upcoming events in a specific calendar.	/private/cs/calendar/getissues	{currentdatetime}/{type}/{limit}	currentdatetime type limit	valid time format INVALID_TYPE = -1 / PRIVATE_TYPE = 0 / SHARED_TYPE = 1 / PUBLIC_TYPE = 2 integer
updateStatus	To update the status of a task.	/private/cs/calendar/updatestatus	{taskid}	taskid	task id
getCalendars	It gets a list of calendars.	/private/cs/calendar/getcalendars			

4.2.2. Mail application

Mail application of eXo Collaboration use MailWebservice to provide all APIs for working with mail as send mail, check mail, store mail to JCR, etc.

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

Service name	Service URL	Description	Location
MailWebservice	<code>\$portalname/\$restcontext/this-service-is-used-to</code>	This service is used to call extended services of Mail application.	- Maven groupId: org.exoplatform.cs - artifactId: exo.cs.web.webservice

- **APIs Usage:**

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

Name	Description	Service URL endpoint	Parameters	Expected Values
checkMail	It executes to check mails when having a request.	<code>/checkmail/{username}/{accountId}/{folderId}/</code>	username accountId folderId	user id account id folder id
synchFolders	It synchronizes the mail folders in the clients and those in the mail server.	<code>/synchfolders/{username}/{accountId}/</code>	username accountId	user id account id
stopCheckMail	to stop checking the mail	<code>/stopcheckmail/{username}/{accountId}/</code>	username accountId	user id account id
getCheckMailJobInfo	It is the method to get information of mail-checking job.	<code>/checkmailjobinfo/{username}/{accountId}/</code>	username accountId	user id account id
searchemail	It is used to search information from emails	<code>/searchemail/{keywords}</code>	keywords	string

4.2.3. Chat application

Chat application use some APIs to help users to create a room, join a room, invite other users to room, send files, etc.

And the chat application of eXo Collaboration uses two services: `RESTXMPPService` and `FileExchangeService` to do these tasks.

4.2.3.1. `RESTXMPPService`

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

Service name	Service URL	Description	Location	
RESTXMPPService	<code>\$portalname/\$restcontextname</code>	It is the service to implement all the action sent to the chat server;	* Maven groupid: <code>org.exoplatform.cs</code> * <code>exo.cs.eXoApplication.chat.service</code>	

- **APIs Usage:**

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

Name	Description	Service URL endpoint	Parameters	Expected Values
loadJsResourceBundle	To read the language files in the chat server.	<code>/loadJsResourceBundle/{locale}/</code>	<code>{locale}</code>	locale id
createRoom	To create a chat room or a group chat	<code>/muc/createroom/{username}/</code>	username room nickname	user id room name display name
configRoom	To establish the configuration of a chat room.	<code>/muc/configroom/{username}/</code>	username room	user id room name
getRoomConfigForm	To get the configuration of a chat room created.	<code>/muc/getroomconfig/{username}/</code>	username room	user id room name
getRoomInfo	To get the information of a	<code>/muc/getroominfo/{username}/</code>	username	user id

Name	Description	Service endpoint	URL	Parameters	Expected Values
	chat room created			room	string
getJoinedRooms	To list chat rooms that a user has been joined.	/muc/joinedrooms/{username}/	{username}		user id
getRooms	To get a list of group chat or chat rooms created.	/muc/rooms/{username}/	{username}		user id
declineToRoom	To refuse the invitation to join the chat room.	/muc/decline/{username}/{inviter}/	{username}	inviter room reason	user id user id room name string
destroyRoom	To delete a chat room created.	/muc/destroy/{username}/	{username}	room reason altroom	user id room name string room id
inviteToRoom	To invites other users to join a chat room.	/muc/invite/{username}/{invitee}/	{username}	invitee room reason	user id user id room name string
joinRoom	To join a chat room.	/muc/join/{username}/	{username}	room nickname password	user id room name display name room password
leftRoom	To leave a chat	/muc/leaveroom/{username}/			

Name	Description	Service endpoint	URL	Parameters	Expected Values
	room.			username	user id
				room	room name
changeNickname	To change the nickname of users	/muc/changenickname/	{username}/{nickname}/	username	user id
				nickname	display name
changeAvailabilityStatusRoom	To change the status of a user in the chat room.	/muc/changesstatus/{username}/	{username}/{mode}/	username	user id
				mood	presence type
				room	room name
				status	presence type
changeSubject	To change the subject of a chat room.	muc/changesubject/{username}/	{username}/	username	user id
				room	room name
				subject	string
manageRoleRoom	To change the role of each user in a chat room.	/muc/managerole/{username}/	{username}/	username	user id
				room	room name
				nickname	display name
				role	Participant / moderator
				command	grant/revoke
manageAffiliationRoom	To change the ownership of a chat room.	/muc/manageaffiliation/	{username}/	{username}/	user id
				room	room name
				nickname	display name

Name	Description	Service endpoint	URL	Parameters	Expected Values
				affiliation	String affiliation
				command	grant / revoke
kickUserFromRoom	To remove a user from the chat room.	/muc/kick/{username}/		username	user id
				nickname	display name
				room	room name
				reason	string
banUserFromRoom	To ban a user in the chat room.	/muc/ban/{username}/		username	user id
				room	room name
				name	user id
				reason	string
addBoddyToRoster	To add a user into the contact list	/roster/add/{username}/		{adduser} username	user id
				adduser	use id
				nickname	display name
				group	group id
updateBoddy	To update new users into the contact list	/roster/update/{username}/		{upduser} username	user id
				upduser	user id
				nickname	display name
				group	group id
createGroup	To create a chat room.	/roster/group/{username}/		{group} username	user id
				group	group id

Name	Description	Service endpoint	URL	Parameters	Expected Values
askForSubscription	To change the presence type of a user into Subscription.	/askforsubscription/	{username}/{askuser}/ username	askuser nickname	user id user id display name
cleanBuddylist	To remove a user from the contact list.	/rosterclean/	{username}/	username	user id
getAllHistory	To get all the chat history of two users.	/history/getmessages/	{username}/{isGroupChat}/ username	isGroupChat usernamefrom	user id true / false user id
getHistoryBetweenDate	To get the chat history of two users in a specific period.	/history/getmessages/	{username}/{isGroupChat}/{from}/{to}/ username	isgroupchat from to usernamefrom	user id true / false valid date format valid date format user id
getHistoryFromDateToNow	To get the chat history of two users from a specific date to the current time.	/history/getmessages/	{username}/{isGroupChat}/{from}/ username	isGroupChat from usernamefrom	user id true / false valid date format valid date format user id
getAllHistoryFile	To download all the chat history file of two users.	/history/file/getmessages/	{username}/{isGroupChat}/{clientTimezoneOffset}/ username	isGroupChat	user id true / false

Name	Description	Service endpoint	URL	Parameters	Expected Values
				clientTimezoneOffset	Long
				usernamefrom	user id
getHistoryFromDateToNowFile	To download the chat history file of two users from a specific date to the current time	/history/file/getmessages/{username}to/{isGroupChat}/{from}/{clientTimezoneOffset}		usernamefrom	user id
				isGroupChat	true / false
				from	valid date format
				clientTimezoneOffset	Long
				usernamefrom	user id
getHistoryBetweenDateFile	To download the chat history file of two users in the specific date.	/history/file/getmessages/{username}to/{isGroupChat}/{from}/{to}/{clientTimezoneOffset}		usernamefrom	user id
				isGroupChat	true / false
				from	valid date format
				to	valid date format
				clientTimezoneOffset	Long
				usernamefrom	user id
getUserInfo	To get the information of a user	/getuserinfo/{username}/{needinfo}/		username	user id
				needinfo	string
login2	To allow a user to log in the chat server.	/login2/{forcache}/		forcache	
logout	To allow a user to log out the chat server	/logout/{username}/{presencestatus}/		username	user id
				presencestatus	presencestatus
messageReceive	To receive a message from other users	/history/messagereceive/{username}/{messageid}/		username	user id
				messageid	message id

Name	Description	Service endpoint	URL	Parameters	Expected Values
removeBuddy	To delete a contact from the contact list.	/roster/del/{username}		removebuddy removebuddy	user id user id
removeTransport	To reset the presence type at the service that is being used.	/removetransport/{username}		transport transport	user id transport serve (e.g: Yahoo, XMPP)
searchUsers	To search users in the chat server	/searchuser/{username}		search search byUsername byName byEmail searchService	user id string true / false true /false true / false string
sendMessage	To send an message to other users.	/sendmessage/{username}		messageBean	usersr id object
sendMUCMessage	To send a message to multile users or a group	/muc/sendmessage/{username}		messageBean	user id object
setUserStatus	To change the status of a user.	/sendstatus/{username}		status	user id available/ unavailabe / do not disturb / away / extend away
subscribeUser	To change presence type into Subcribed type	/subscribeuser/{username}		subsuser	user id user id

Name	Description	Service endpoint	URL	Parameters	Expected Values
unsubscribeUser	To change presence type into the Unsubscribed type	/unsubscribeuser/{username}/{unsubsuser}/	username	unsubsuser	user id user id
acceptFile	To accept getting a file sent from another user.	/fileexchange/accept/{username}/{uuid}/	username	uuid	user id string
rejectFile	To refuse getting a file sent from another user.	/fileexchange/reject/{username}/{uuid}/	username	uuid	user id string
getPreviousStatus	To get the tatus of a user in the last log-in.	/getprevstatus/{username}/	username		user id

4.2.3.2. FileExchangeService

REST API *FileExchangeService* for upload file is defined in `org.exoplatform.services.xmpp.rest.FileExchangeService`

Service name	Service URL	Description	Location
FileExchangeService	<code>\$portalname/\$restcontextname/fileexchange</code>	To upload a file to the server and inform the user that he can download it to his local computer.	- Maven groupId: <code>org.exoplatform.cs</code> - InterfactId: <code>exo.cs.eXoApplication.chat.service</code>

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

Name	Description	Service endpoint	URL	Parameters	Expected Values
upload	To upload a file to the server.	<code>\$portalname/\$restcontextname/fileexchange</code>	description	username requestor	string user id user id

Name	Description	Service endpoint	URL	Parameters	Expected Values
				isroom	true / false