How to install Openfire using Tiki for its userlist

Openfire is a real time collaboration (RTC) server supporting XMPP (Jabber) and WebRTC. See also Why Openfire

This page describes how an instance of Openfire can be set up in such a way that it will make use of the Tiki database for:

- Users
- Groups
- Authentication

Note that this is a specialized setup, that has various drawbacks.

Preparation / Assumptions

It's assumed that the instance of Tiki with which is to be integrated is available. This guide applies only to Tiki instances that use their own database tables to store users, groups and permissions. This guide has not been tested with (and is not expected to work for) instances that are configured to integrate by means of Shibboleth, PHPBB, LDAP, etc.

Openfire will need to be able to query Tiki's SQL database. A database account should be available that can perform all queries below. Note that permissions can be limited to SELECT queries. When Openfire is not installed on the same physical server as Tiki, ensure that firewall and database connectivity is configured to allow access.

Before continuing, ensure that you have access to an account that has administrative permissions in Tiki. Such an account is needed to administer Openfire during the setup.

Install Openfire

Obtain an installer for Openfire, preferably from https://www.igniterealtime.org/. This guide is applicable to any platform, and any version of Openfire (assuming it is somewhat recent).

After downloading, execute the installer, and install Openfire using its web-based admin console. Even though Openfire will (at a later stage), be configured to use Tiki's database, it does need its own database (to store non-user related content). Feel free to use either the standard, or embedded database. If you're using a standard database, you can use the same database as Tiki (Openfire table names do not clash with Tiki's).

Configuring Openfire with "Default" profile settings, meaning that it will use its database for user and groups (we'll change this later).

In the last step of the setup procedure, you'll be asked to create and admin account. This account will be used when reconfiguring Openfire to make use of Tiki's database, later. Afterwards, it will no longer be used, but as it lingers in the database, it's advisable to use properly secure credentials.

After installation, verify that you can log into the Openfire admin console, using the credentials that you created in the last step of the setup.

Reconfigure Openfire to make use of Tiki's database.

In the Openfire admin console, open Server > Server Manager > System properties. There, delete the following properties, if they're defined:

- provider.admin.className
- provider.auth.className
- provider.group.className
- provider.user.className

Next, shut down the Openfire process completely.

After Openfire is shut down, find the file named openfire.xml in Openfire's 'conf' directory.

Add the following snippet of XML as a direct child of of the 'jive' root element, making sure to replace the DBUSERNAME and DBPASSWORD placehodlers in the first few lines with actual database credentials.

openfire.xml configuration

```
<jdbcProvider>
  <driver>com.mysql.jdbc.Driver</driver>
<connectionString>jdbc:mysql://wikisuite.net/trunk?user=DBUSERNAME&amp;password=DBPASSWORD</connectionString>
</jdbcProvider>
<provider>
  <user>
    <className>org.jivesoftware.openfire.user.JDBCUserProvider</className>
  </user>
  <group>
    <className>org.jivesoftware.openfire.group.JDBCGroupProvider</className>
  </group>
  <admin>
    <className>org.jivesoftware.openfire.admin.JDBCAdminProvider</className>
  </admin>
  <auth>
    <className>org.jivesoftware.openfire.auth.JDBCAuthProvider</className >
  </auth>
</provider>
<jdbcUserProvider>
  <loadUserSQL>SELECT value, email FROM users_users uu LEFT JOIN tiki_user_preferences up ON (uu.login = up.user AND up.prefName LIKE 'realName') WHERE login=?</loadUserSQL>
  <userCountSQL>SELECT COUNT(*) FROM users_users</userCountSQL>
  <allUsersSQL>SELECT login FROM users_users</allUsersSQL>
  <searchSQL>SELECT login FROM users_users uu LEFT JOIN tiki_user_preferences up ON (uu.login = up.user AND up.prefName LIKE 'realName') WHERE</searchSQL>
  <usernameField>login</usernameField>
  <nameField>value</nameField>
  <emailField>email</emailField>
</jdbcUserProvider>
<jdbcGroupProvider>
     <groupCountSQL>SELECT count(*) FROM users_groups</groupCountSQL>
     <allGroupsSQL>SELECT groupName FROM users_groups</allGroupsSQL>
     <userGroupsSQL>SELECT DISTINCT groupName FROM users_usergroups ug JOIN users_users uu ON (ug.userId = uu.userId AND login = ?)</userGroupsSQL>
     <descriptionSQL>SELECT groupDesc FROM users_groups WHERE groupName=?</descriptionSQL>
     <loadMembersSQL>SELECT login FROM users_usergroups ug JOIN users_users uu ON (ug.userId = uu.userId AND ug.groupName = ?)</loadMembersSQL>
     <loadAdminsSQL>SELECT login FROM users_usergroups ug JOIN users_users uu ON ug.userId = uu.userId WHERE ug.groupName = ? AND EXISTS(SELECT userId FROM users_usergroups WHERE userId = ug.userId and groupName = 'Admins')</loadAdminsSQL>
</jdbcGroupProvider>
<jdbcAdminProvider>
  <getAdminsSQL>SELECT login FROM users_grouppermissions ugp JOIN users_usergroups ug JOIN users_users uu ON ugp.groupName = ug.groupName AND ug.userId = uu.userId AND ugp.permName = 'tiki_p_admin'</getAdminsSQL>
</jdbcAdminProvider>
<jdbcAuthProvider>
  <passwordSQL>SELECT hash FROM users_users WHERE login = ?</passwordSQL>
  <passwordType>bcrypt</passwordType>
  <allowUpdate>false</allowUpdate>
  <bcrypt>
    <cost>10</cost>
  </bcrypt>
</jdbcAuthProvider>
```

Save the file and restart Openfire.

After Openfire is started, re-open the file. The snippet that you added should have been removed by Openfire. If it did not, or if smaller fragments remain, then you likely neglected to delete some of the properties mentioned above. If you can still log into Openfire, you can try to delete the properties that relate to the fragments still in the file, and restart Openfire again. If this does not work (or if you can't login anymore), it's probably easiest to remove Openfire completely, and start over.

Wrapping up

Try logging in to the Openfire admin console again, but this time, use a Tiki administrative account, instead of the Openfire admin account! If you're able to log in, it's pretty likely that you were successful.

To verify that things are working as expected, review the users and groups in Openfire (using the "Users/Groups" tab in the admin console). These should now list the Tiki users and groups!

Drawbacks / Restrictions

The approach described in this guide depends on the fact that PHP uses brypt to encrypt the Tiki user hashes (which is a default, but can be configured differently, depending on the Tiki, PHP and OS setup).

For authentication to succeed, XMPP clients will need to support the PLAIN mechanism for SASL. This mechanism is commonly supported, but is regarded as the least secure mechanism to be used. Notably, this configuration will not allow clients to use the SCRAM-SHA-1 mechanism, which the XMPP specification define as a mandatory-to-be-supported mechanism.

Renaming a user in Tiki will be more or less like creating a new user. Nothing will be kept in Openfire (message history, contacts, etc.)