ELIXIR AMBIENCE

AMBIENCE / REPERTOIRE ADMIN GUIDE

Table of Contents

Ambience / Repertoire Administrator Guide

1. Introduction

This guide describes the tasks that administrators may need to perform on Ambience/Repertoire software suite.

The tasks of an administrator may include the following:

- Deploy licence
- Set up roles and users
  - Create identity
  - Create role
  - Create user
  - Assign privileges
- Set up email server
- Set up authentication
  - External authentication
  - Time-based One-time Password (TOTP) Two-factor Authentication (2FA) (see page)
- Configure Horizontal Scalability using Nginx

The following sections uses Ambience as example.

Refer to the following websites for more information:

- https://docs.elixirtech.com/Ambience/2025.0/index.html
- https://docs.elixirtech.com/Repertoire/2025.0/index.html
- www.elixirtech.com
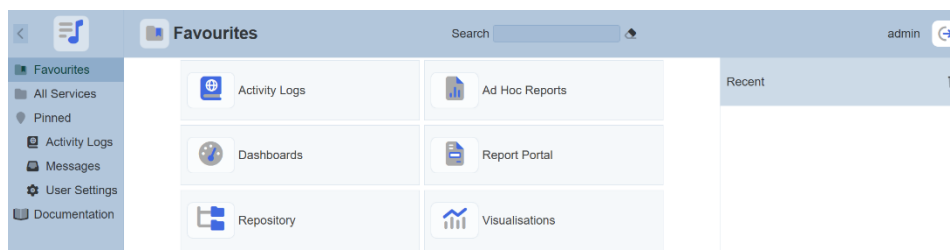
2. Deploy Licence

When the Ambience/Repertoire software suite is initially installed, it comes with a default licence. This default licence allows a minimal set of modules and does not have an expiration date.

To access to other modules in Ambience software, a new licence needs to be imported into the software. This new licence has an expiration date and need to be replaced before the expiry, or the software will fall back to the minimal set of modules.

There are two methods to import the licence, either by using the software interface or by using the command line interface. This method is useful during setup as you can install the licence before starting the server, hence avoiding the need to stop and restart.

To import a licence using the software interface:

1. From the "Service Chooser" page, select the "Licence Deployer" in the right panel. If Ambience is initially installed and logged in for the first time, step 2 will appear directly.


2. The "Licence Deployer" page will appear.
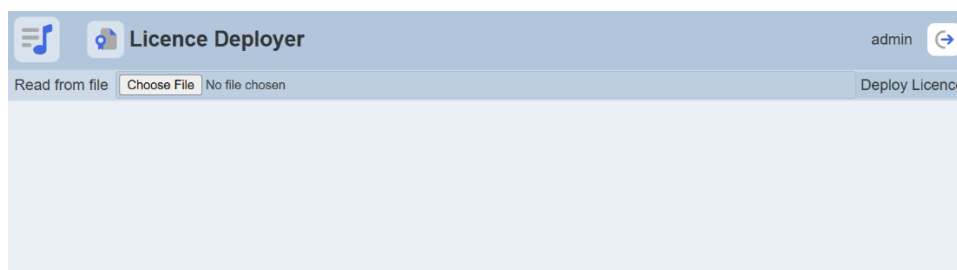
3. Click on the "Choose File" button and browse to the location of the license file (*ElixirAmbience.licence* for Ambience or *ElixirRepertoire.licence* for Repertoire). The content of the licence file will appear in the editor.

4. If the content is correct, click on the "Deploy Licence" button at the upper right corner of the page.

5. Upon successful upload, a message will appear.
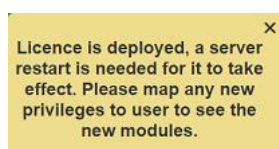
6. After the licence has been deployed to the server. Restart the server for the licence to take effect.

7. Ensure to map any new privileges to the desired users to see the new modules.

To import a licence using the command line interface:

- Open a terminal window and navigate to the software main folder, then navigate to the "bin" folder.

- In the command line, key in the following command:

  For Ambience:

  ```
  ambience-cli deploy-licence <path of ElixirAmbience.licence file>
  ```
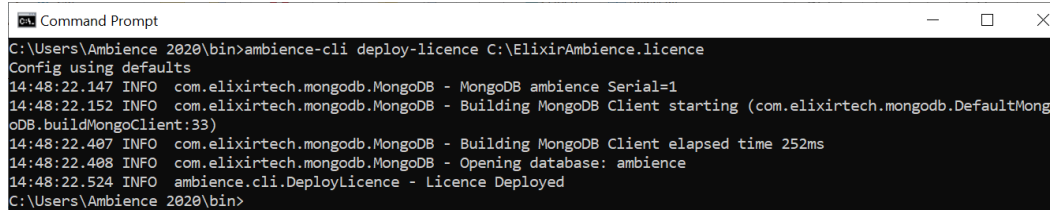
  For Repertoire:

  ```
  ambience-cli deploy-licence <path of ElixirREpertoire.licence file>
  ```

- The command line window will display messages indicating the successful deployment.

```
C:\Users\Ambience 2020\bin>ambience-cli deploy-licence C:\ElixirAmbience.licence
Config using defaults
14:48:22.147 INFO  com.elixirtech.mongodb.MongoDB - MongoDB ambience Serial=1
14:48:22.152 INFO  com.elixirtech.mongodb.MongoDB - Building MongoDB Client starting (com.elixirtech.mongodb.DefaultMong
oDB.buildMongoClient:33)
14:48:22.407 INFO  com.elixirtech.mongodb.MongoDB - Building MongoDB Client elapsed time 252ms
14:48:22.408 INFO  com.elixirtech.mongodb.MongoDB - Opening database: ambience
14:48:22.524 INFO  ambience.cli.DeployLicence - Licence Deployed
C:\Users\Ambience 2020\bin>
```

- Restart the server to allow the new licence to take effect.

3. Setup Roles and Users

In the software, privileges are access rights to a particular module or functionality. A role is a set of privileges customized to perform a certain function. Roles or extra privileges are assigned to users. End-users will see roles but not privileges.

The administrator can manage the users and roles using the following modules in the software:

- Identities
- Users
- Roles
- Privileges

Below is the workflow for creating a user.
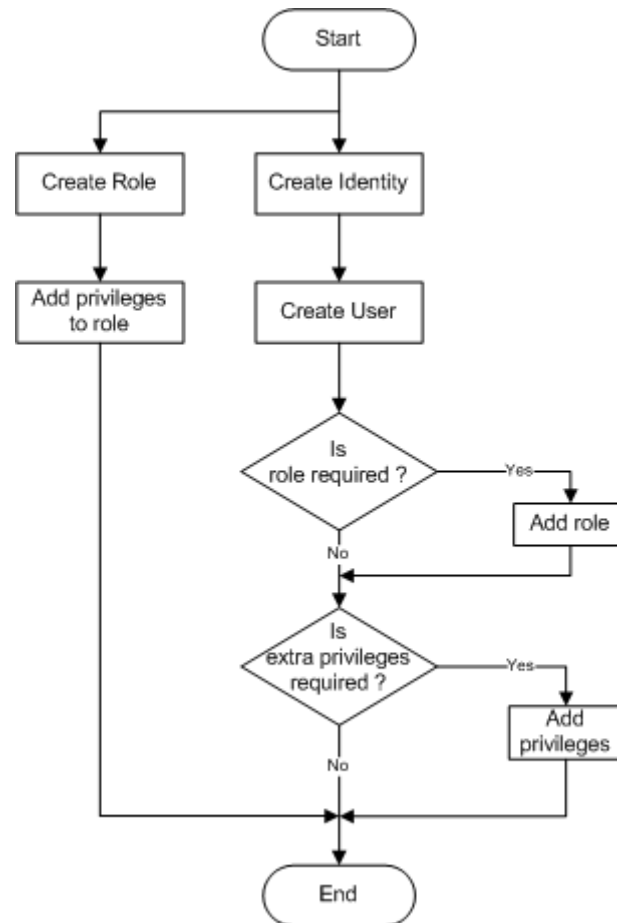
3.1. Privileges

The Privileges module provides a simple interface to manage privileges.

Privileges are access rights to interface modules. By default, privileges used by all standard modules are listed. If there are custom modules, adding privileges used by those modules is to be done through this interface. Hence, most users will not need access to this module, except technical users and system administrators.

Modules require certain privileges to be present, for example, *mod-dashboard*. If you delete *mod-dashboard* then nobody will be able to view dashboards. Therefore, it is not advisable to rename or delete any existing standard ones, only to create or manage new custom ones. Ambience modules will automatically recreate any required privileges when the server is next started. So, if a privilege is accidentally deleted (and too late to undo), you can recover by restarting the server. It is also a good idea to prefix custom modules with a unique prefix, for example, *abc-custom-privilege*, rather the use *mod-* to avoid clashes if future Ambience releases include new *-mod-* privileges.
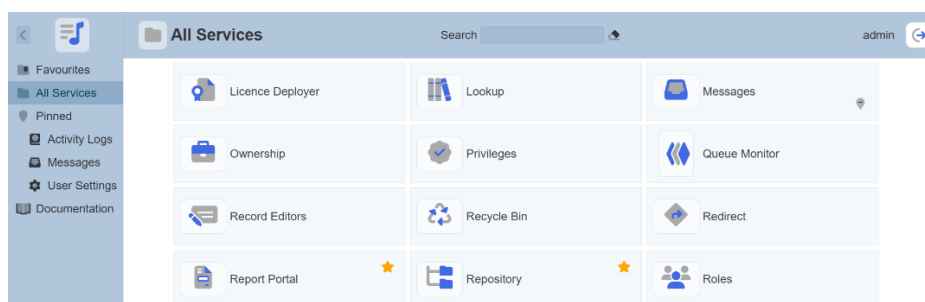
This module allows you to add, edit and delete privileges.

3.1.1. Add Privilege
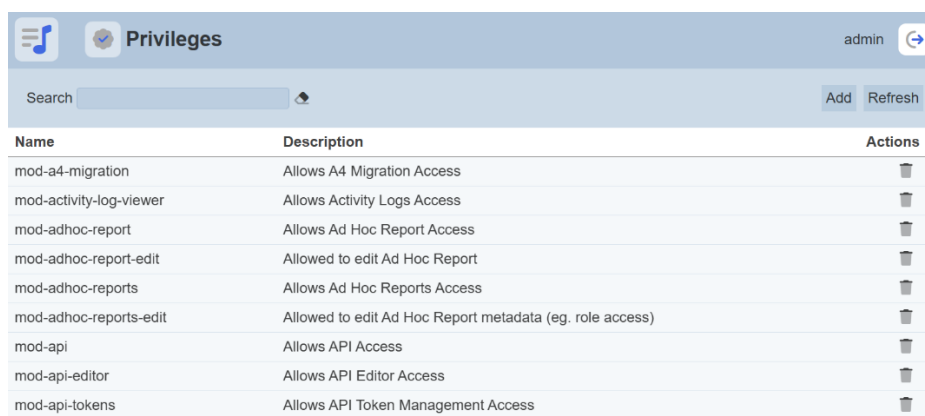
Use the following steps to add a privilege:

1. Click on the "Elixir Ambience" icon on the upper left corner of the page and select "All Services".
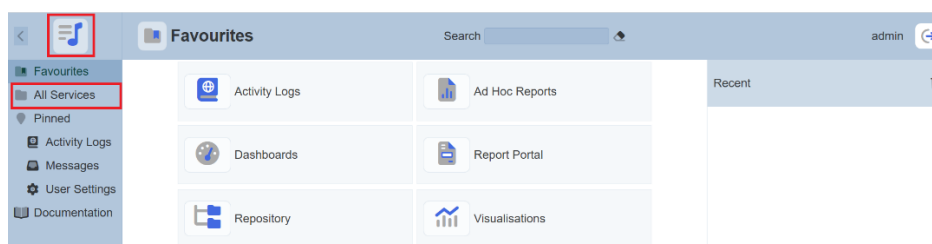

2. the "All Services" page, select "Privileges" in the right panel.
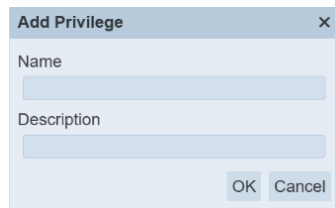
3. The "Privileges" page will appear.

4. Click on the "Add" button on the upper right corner of the page to add a new privilege.
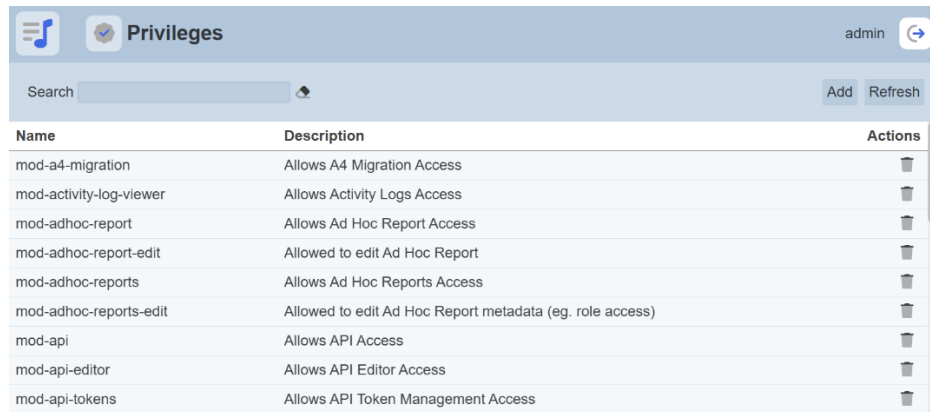
5. In the "Add Privilege" dialog box, key in the following:

   - "Name" field   – Key in a unique name for the new privilege
   - "Description" field – Key in a brief description of the new privilege

6. Click on the "OK" button to add the new privilege.

3.1.2. Edit Privilege
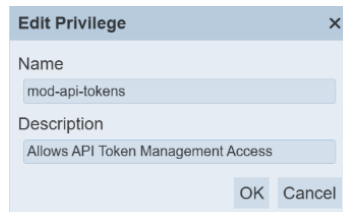
Use the following steps to add a privilege:

1. In the "Privileges" page, click on the name of the privilege under the "Name" column.

2. The "Edit Privilege" dialog box will appear.

3. Make the necessary change and click on the "OK" button to save the changes.

3.1.3. Delete Privilege

To delete a privilege, click on the 🗑 "Delete" icon under the "Actions" column corresponding to the desired privilege.

There is an option to undo the deletion. A notification with an "Undo" button appears after clicking on the "Delete" icon.
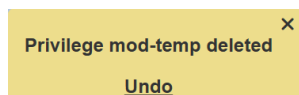
Upon clicking on the "Undo" button, the deleted user is restored and is added back to the list of privileges.

3.2. Roles

The Roles module provides a simple interface to manage user roles. Privileges or access rights to the different modules can be grouped together into roles. Adding a user to a role collectively grants the privileges to the said user. This is particularly useful and makes it easy to maintain when there are two or more users requiring the same set of privileges.

The "Least Privilege" mindset should be considered from a server hardening point of view. It is best to define roles with the least privilege to do the job. For ease of setup and evaluation, the initial admin user is given all privileges, but this should be limited in any real-world deployment. However, ensure that one user does have permission to access Roles and Users modules, else it is possible to "lock yourself out" when nobody has sufficient privilege to configure roles.

This module allows you to add, edit and delete roles.

3.2.1. Add Role

Use the following steps to add a new role:

1. Click on the 🎵 "Elixir Ambience" icon on the upper left corner of the page and select the "All Services".
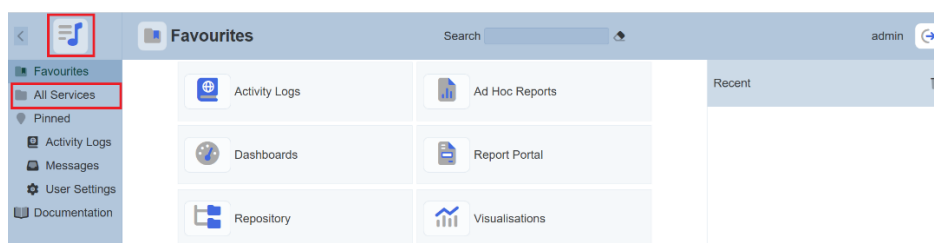
2. In the "All Services" page, select "Roles" in the right panel.
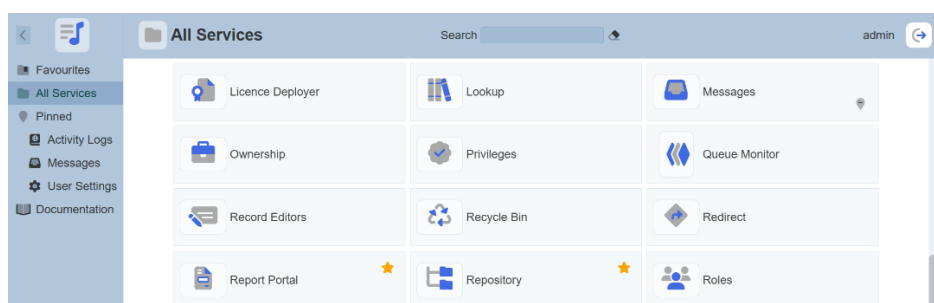
3. The "Roles" page will appear.
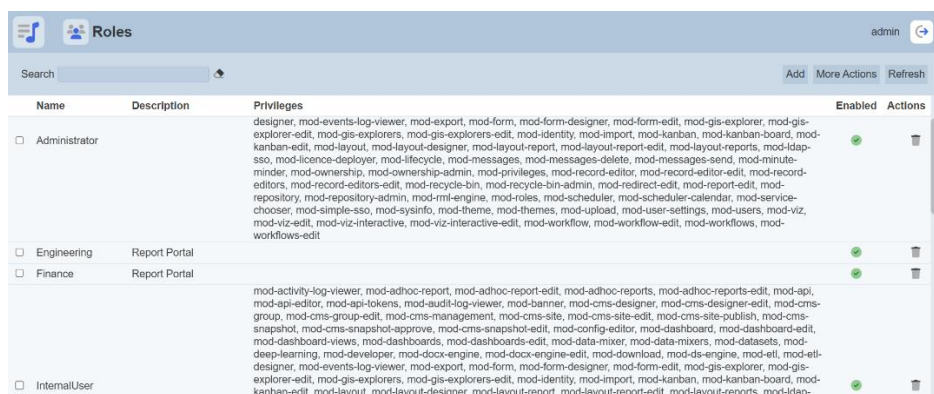
4. Click on the "Add" button on the upper right corner of the page to add a new role. The "Add Role" dialog box will appear.

5. In the dialog box, key in and select the following:

- "Name" field   – Key in a unique name for the new role
- "Description" field   – Key in a brief description for the new role
- "Privileges" tab   – Selects the desired privileges for the new role
- "Users with this Role" tab   – Selects the desired users for the new role

6. In the two tabs, you can use the search function at the top of each respectively section to search for the desired privilege or users. You can also use the "Select All", "Select None" and the "Invert Selection" icons to aid your selection.

7. Click on the "OK" button e to add the new role.

3.2.2. Edit Role

Use the following steps to edit a role:

1. In the "Roles" page, click on the name of the role to edit under the "Name" column.

2. The "Edit Role" dialog box will appear.

3. Make the necessary changes and click on the "OK" button to save the changes.

3.2.3. Delete Role

To delete a role, click on the 🗑 "Delete" icon under the "Actions" column corresponding to the desired role.

There is an option to undo the deletion. A notification with an "Undo" button appears right after clicking on the "Delete" icon.
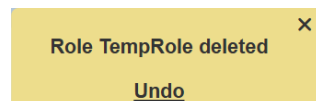
Upon clicking on the "Undo" button, the deleted user is restored and is added back to the list of roles.

3.3. Identity

People need to be identified (authenticated) before they can be authorised to use all the different modules in the software suite. If you already have an authentication system, such as an SSO, LDAP or Active Directory, then it is possible to use that as the authentication mechanism. However, if you do not have a mechanism, or want to keep the software separate, an identity module is provided which handles the authentication part of the login process. This identity management system is built upon OAuth2, which is what makes it possible to plug in alternate authentication providers.

The Identities module provides a simple mechanism for authentication (determining who is logging in). The Users module provides the corresponding mechanism for authorisation (determining what each authenticated user can access).

The Identities module provides a simple interface to manage user identities. Once the user identity has been created, it can be assigned roles and privileges through the Users module.

The functions of the Identities module are:

- Add identity
- Upload a list of identities
- Enable/disable identity
- Edit details of identity
- Reset password of an identity

3.3.1. Add Identity

Use the following steps to add an identity:
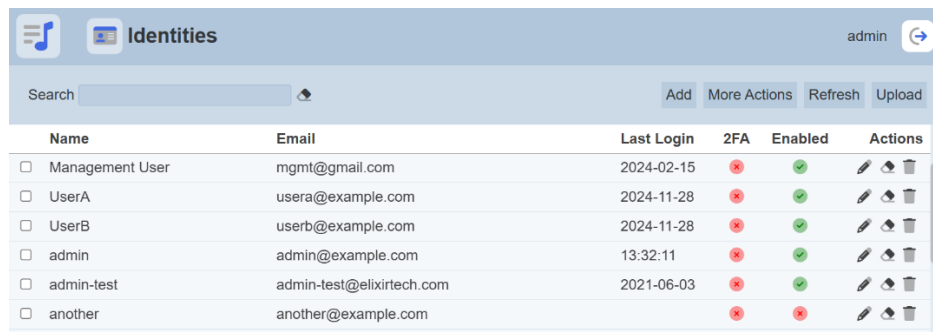
1. Click on the "Elixir Ambience" icon on the upper left corner of the page and select the "All Services".
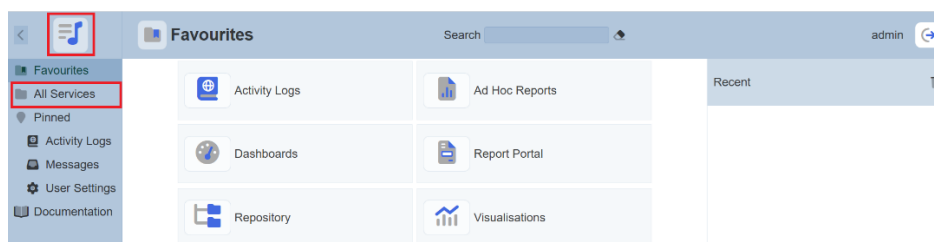
2. In the "Service Chooser" page, select "Identities" in the right panel.
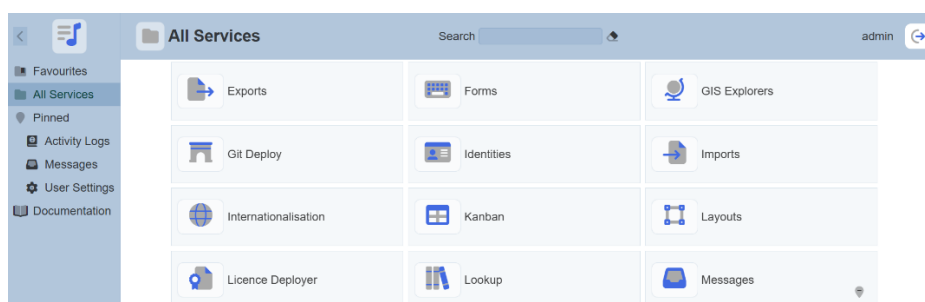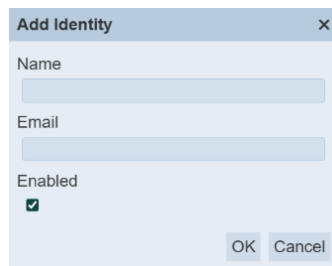
3. The "Identities" page will appear.

4. To add a new identity, click on the "Add" button at the upper right corner of the page. The "Add Identity" dialog box will appear.
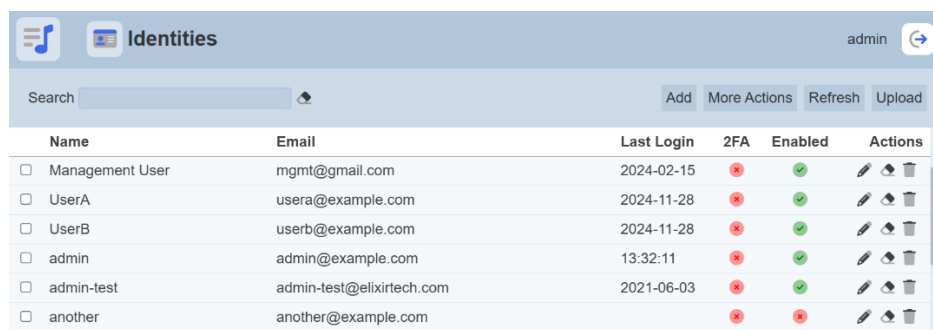
5. In the dialog box, key in the following:

- "Name" field   – Key in a unique name
- "Email" field   – Key in an email of the new identity
- "Enabled" field – Ensure it is selected

6. Click on the "OK" button to add the new identity.

3.3.2. Upload Identity

Instead of adding users one at a time, you can upload a list of users using a file.
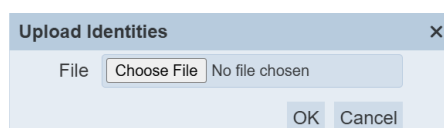
Use the following steps to upload identities:

1. In the "Identities" page, click on the "Upload" button at the upper right corner of the page.

2. The "Upload Identities" dialog box will appear.

3. Browse to the location of the desired file and click on the "OK" button.

4.   Upon successfully uploading, a notification will appear.
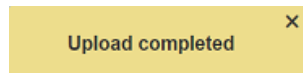
5.   If there is any conflict during uploading, for example, identity already exist, an error message will appear and the action will be aborted.
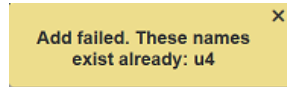
3.3.3. Edit Identity

The "Edit" function allows you to change the details of the identity.

Use the following steps to edit an identity:

1.   In the "Identities" page, click on the name of the identity to edit under the "Name" column.

2.   The "Edit Identity" dialog box will appear.

3.   Make the necessary changes to the identity. You can also enable or disable the identity in this dialog box.

4.   Click on the "OK" button to save the changes.

3.3.4. Delete Identity

You can remove a user from Ambience software by clicking on the 🗑 "Delete" icon under the "Actions" column corresponding to the desired user.

There is an option to undo the deletion. A notification with an "Undo" button appears right after clicking on the "Delete" icon.
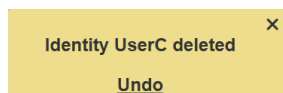
Upon clicking on the "Undo" button, the deleted user is restored and is added back to the list of users.

3.3.5. Reset Password
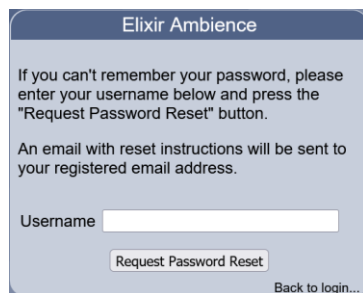
There are two ways to reset a user password. One is during login and the other using the Identities module.

Use the following steps to reset password using login:

1. In the login dialog box, click on the "Can't access your account?" link at the bottom right of the dialog box.

2. Key in the user name and click on the "Request Password Reset" button. To abort the action, click on the "Back to login…" link at the bottom right.

3. A notification will appear notifying the user of the change.
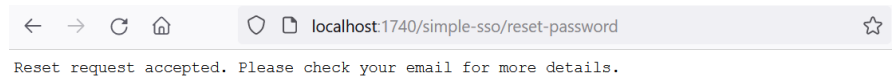
Use the following steps to reset password using Identities module:

1. In the "Identities" page, click on the ⬦ "Reset Password" icon under the "Actions" column corresponding to the desired identity.

2. A notification will appear to inform you that the password has been reset.
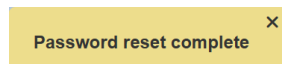
3. If the email server has been set up, an email will be sent to the user notifying them of the password reset and supplying them with the new randomly generated password.

When the user logs in with the randomly generated password, they will be forced to change the password immediately. This can be disabled by editing the setting in the Ambience "etc" folder, in the *application.conf* file.

```
elixir.Identity {
  ...
  on-reset {
    changePassword = false
  }
}
```

3.3.6. Reset 2FA

If 2FA has been enabled and setup in the software, you can reset the 2FA of a user using the Identity module by clicking on the "Reset 2FA" icon under the "Actions" column corresponding to the desired user.

There is no undo for this action. If the user will need to set up the 2FA again.

3.4. User

The Users module provides a simple interface to manage user authorization.

Privileges or access rights to the different modules can be granted to a user in several ways:

- Adding the user to a role that has those privileges
- Adding those privileges directly to the user's set of extra privileges
- Both of the mentioned options

Privileges added through roles and extra privileges are both read in. For instance, if a user's role only has the mod-dashboard privilege but the user has the *mod-dashboard-edit* privilege as an extra privilege, the user would benefit from both privileges.

Users must be added through this module first before privileges can be granted. Suspending user access can also be done by disabling users. This action can easily be undone by enabling disabled users.

Another option for assigning roles to existing users is to use the Roles module. There is no difference in the effect of role assignments between assigning roles to users through the Users module and assigning users to roles through the Roles module.

The authentication (or login process) is managed separately (e.g. Identities module or externally). The name assigned to the user through the Users module should match the username used by the authentication mechanism to be able to successfully authorize the user.

The functions of the Users module are:

- Add user
- Edit user
  - o Change user's name
  - o Enable/disable user
  - o Add/remove role
  - o Add/remove extra privilege
- Delete user

3.4.1. Add User

Use the following steps to add a user:

1. Click on the ♪ "Elixir Ambience" icon on the upper left corner of the page and select the "All Services".
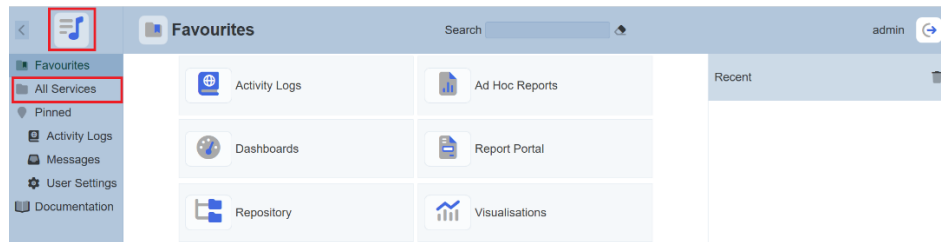
2. In the "Service Chooser" page, select "Users" in the right panel.
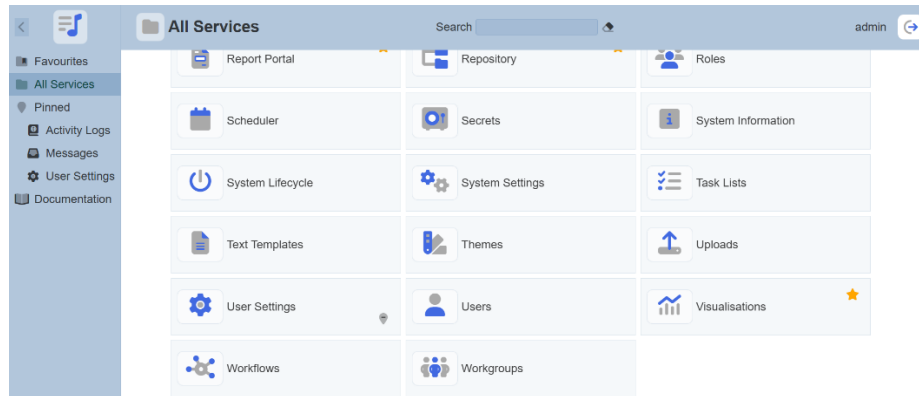
3. The "Users" page will appear.
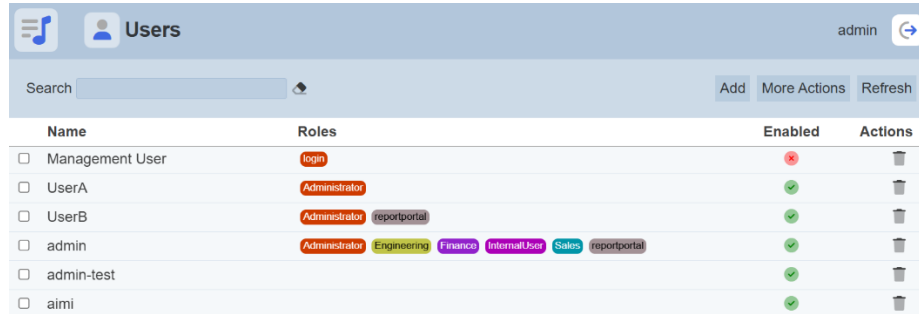
4. In the "Users" page, click on the "Add" button on the upper right corner of the page. The "Add User" dialog box will appear.
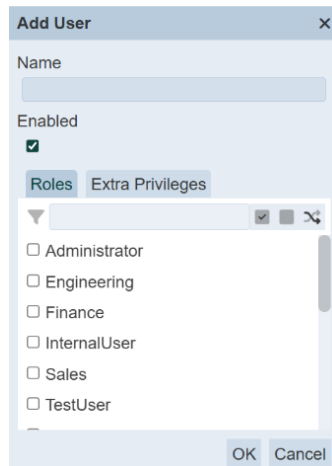
5.   In the dialog box, key in and select the following:

- "Name" field   – Key in the username created in the Identities module
- "Enabled" field   – Ensure this field is selected
- "Roles" tab   – Selects the appropriate role(s)
- "Extra Privileges" tab – Selects the appropriate privilege(s)

By assigning a particular role(s) to the user, the user will inherit all the privileges assigned to the role.

If there is any privilege that the user requires but does not fall in the role assigned to the user, the administrator can add extra privileges to the user by using the "Extra Privileges" field. Use this sparingly.

6.   In the two tabs, you can use the search function at the top of each respectively section to search for the desired privilege or users. You can also use the "Select All", "Select None" and the "Invert Selection" icons to aid your selection.

7.   The newly created user will be listed in the "Users" page.

8.   If the email server has been set up, an email will be sent to the new user with a randomly generated password. The new user will need to change the password upon login.
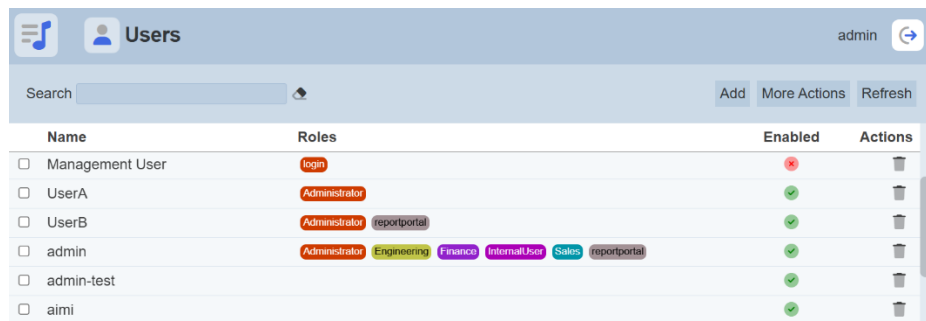
If the email server has not been set up, the random password can be found in a text file in the "/mail" folder in the root directory.

3.4.2. Edit User

The administrator can edit the roles and privileges assigned to the user.
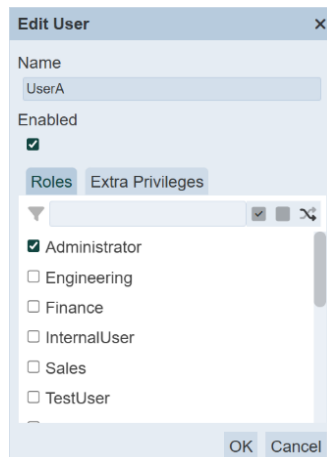
Use the following steps to edit users:

1.   In the "Users" page, click on the name of the user to edit under the "Name" column.

2.   The "Edit User" dialog box will appear.

3. Make the necessary changes and click on the "OK" button.

3.4.3. Delete User

You can remove a user from the software by clicking on the 🗑 "Delete" icon under the "Actions" column corresponding to the desired user.

There is an option to undo the deletion. A notification with an "Undo" button appears right after clicking on the "Delete" icon.
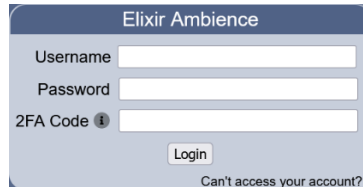
User UserC deleted ✕
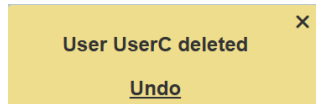
Undo

Upon clicking on the "Undo" button, the deleted user is restored and is added back to the list of users.

3.4.4. Change Password

Normal users will be prompted to change password on their first log in.

In the "Change Password" section, key in the old password and key in the new password twice. Hit the "Request Password Change" button. A verification panel will appear to request the user to key in the validation code.

A validation code will be sent to the user's email upon successful change password request. Key in the validation code and click on the "Verify Password Change" button. A message will appear to inform the user that the password change is successful.
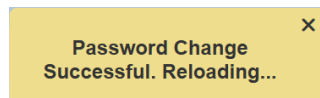
After reloading, the Service Chooser page will appear.

Two things to note:

1. If the "*mod-no-force-password-change*" privilege is granted to the user, the user does not need to change the password. Thus the "Change Password Required" screen will not appear when the user log in for the first time.
2. The validation code sent via email can be turned off in the "*application.conf*" file. This means that the password will be changes with no further verification when the user changes his/hers password. Refer to Application Config File Guide for more information.

4. Email Server and Authentication

4.1. Configure and Test Mail Server

When identity is added, an email is sent with randomly generated password to the user. When a user wants to change the email or password, a verification is sent via email as well.

If you have not set up an email server, the default behaviour is to store the emails in the "/mail" folder within the software. This is usually for diagnosis or debugging purposes. It is recommended to set up a mail server at the start.

Below are two examples of how to set up a mail server.

Example 1: Uses Gmail

1. Gmail allows only OAuth2 authentication without weakening security. Visit https://console.developers.google.com/apis/credentails to set up a "*clientID*" and "*clientSecret*". Use these to generate a "*refreshToken*".

2. In the software root folder, navigate to the "/etc" folder. Open the *application.conf* file using a text editor. In the "*elixir.mail*" section, edit the following with the information obtained earlier accordingly.

```
elixir.mail {
  smtp = "gmail"
  gmail {
    host = "smtp.gmail.com"
    port = 587
    debug = true
    oauth2 {
      userName = "xxx@gmail.com"
      clientId = "XXXX"
      clientSecret = "YYYY"
      refreshToken = "ZZZZ"
    }
  }
}
```

3. After the above is edited in the *application.conf* file, start the software server and go to Identities module to create a user with a valid email address.

Example 2: Uses AWS

1. In the software root folder, navigate to the "/etc" folder. Open the *application.conf* file using a text editor. In the "*elixir.mail*" section, edit the following:

```
elixir.mail {
  smtp = "aws"
  aws {
    from = "<user@example.com>"
    host = "<hostname>"
    dnsResolver = ""
    port = 465
    user = "XXXX"
    password = "YYYY"
    connectionTimeout = 30000
    tls = true
    ssl = true
    authMechanism = ""
    debug = false
  }
}
```

2. After the above is edited in the *application.conf* file, start the software server and go to Identities module to create a user with a valid email address.

4.2. Use GitLab As Authentication

The Identities module in the software provides a simple mechanism for authentication (determining who is logging in). If you already have an authentication system, such as an SSO, LDAP or Active Directory, then it is possible to use that as the authentication mechanism. This identity management system is built upon OAuth2, which is what makes it possible to plug in alternate authentication providers.

If an external authentication system is used, the Identities module is not needed and should be removed to avoid confusion.

This section describes the steps to set up GitLab as the authentication method to log into the software.

The steps are as follows:

1. Create an account in GitLab.

2. In GitLab, add Ambience software as an application under your user.

   Note that URL callback should be http://hostname:1740/authclient for Ambience. Use port 1730 for Repertoire. This is consistent with the setting in *application.conf* file.

3. Change the hostname on your machine to point to the proper endpoint (i.e., the added application).

4. Add the user into Users module with the same name that was created in the GitLab server.

5. Go to the software root directory and go to the "/etc" folder. Open the *application.conf* file using a text editor.

6. Make the following changes in the "*elixir.sso.client*" section.

```
elixir.sso.client {
  cookie-name = "elx-amb"

  cookie-same-site = "Lax"
  openid-field = "name"
  openid-scope = "openid email"
  service-definition {
    elxsso {
      authorization = "https://<gitlab-host>/oauth/authorize"
      token = "https://<gitlab-host>/oauth/token"
      userinfo = "https://<gitlab-host>/oauth/userinfo"
      logout = ${sso-server-baseurl}"/simple-sso/logout"
      debug = false
      client {
        id = "[Your Application ID]"
        secret = "[Your secret]"
        endpoint = ${sso-client-baseurl}"/authclient"
      }
    }
  }
}
```

7. Save the *application.conf* file.

8. Restart the software server. Open a browser and key in "localhost:1740" in the address bar and hit the enter key.

   For Repertoire, key in "localhost:1730" in the address bar.

9. Log into the software with your GitLab account.

4.3. Two-factor Authentication

Ambience/Repertoire software supports Time-based One-time Password (TOTP) Two-factor Authentication (2FA). By default, 2FA is disabled in the *application.conf* file. To enable 2FA, edit the *application.conf* file in two areas:

1. Under the *simple-server* section, change *show-totp = false* to *true*. This is to allow the login dialog to include 2FA.

```
simple-server {
  clients {
    ambience {
      secret = "171ccf22-670a-43c2-ac79-05c44bf305e3"
      redirect = ${sso-client-baseurl}"/authclient"
      #login-page = "" # set resource file here to use a custom
login page for this client
      landing-page = "http://"${host}":"${port}"/"
      name = "Elixir Ambience"
      show-totp = true
    }
  }
}
```

2. Add a new line in the *application.conf* file. This will allow User Settings module to include 2FA setup, in which users can set up their own 2FA.

```
ambience.user-settings.enable-panel.totp = true
```

5. Configure Ambience

5.1. Horizontal Scalability – Nginx

Horizontal scalability is the ability to increase capacity by connecting multiple hardware or software entities so that they work as a single logical unit.

This section describes the steps to set up horizontal scalability (two web servers and more) for Ambience server.

1. Nginx server has been created and configured and running. Its configuration is as follows:

   The path for the configuration for Ubuntu is "etc/nginx/conf.d/nginx.conf".

   Below is an example of the *nginx,conf* file: For Repertoire, change the port to 1730.

   ```
   upstream backend {
     server xxx.xxx.xxx.xxx:1740;
     server xxx.xxx.xxx.xxx:1740;
   }


   server {
     server_name xxx.xxx.xxx.xxx;
     location / {
       proxy_redirect    off;
       proxy_set_header  X-Real-IP $remote_addr;
       proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
       proxy_set_header  Host $http_host;
       proxy_pass        http://backend;
     }
   }
   ```

2. Minimum two or more software servers has been created and configured. Set the Nginx IP and port into each software server's *application.conf* file.

   ```
   ...
   sso-server-baseurl = "http://"${host}":"${port}
   sso-client-baseurl = "http://"${host}":"${port}


   elixirtech.simple-identity.client-landing-page =
   "http://"${host}":"${port}"/index.html"
   ...
   ambience.web {url = "http://"${host}":"${port} ......}
   ```

3. Open two terminals or more on different machines.

4. Go to folder where the "*elx-stub.jar*" file is and run the following command for each terminal:

   ```
   java -Dvisualvm.display.name=Ambience -Djava.awt.headless=true
   -Dlogback.configurationFile=etc/logback.xml -jar elx-stub.jar
   ambience.module.Launcher
   ```

   The software servers are successfully brought up.

5. Open a browser and set the URL as per "ambience.web" configured in step 2. You should be able to open the software portal.

6. Perform some actions in the browser, such as, edit a dashboard, refresh a page, etc.

7. Monitor each software server's log in the Event Logs module. The log for each terminal shows the log in rotation between one and the others. (For Ambience only).

5.2. Deployment

5.2.1. Simple Deployment

For simple deployment, a single host is used. Below is an example using Ambience.
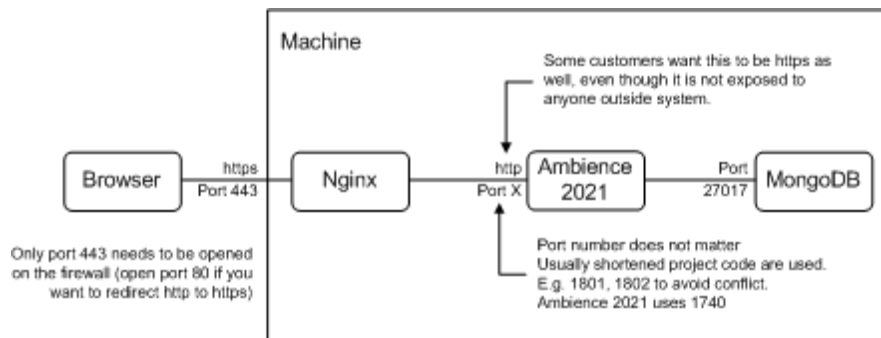
Below is a sample Nginx configuration.

```
server {
  listen 442 ssl;
  server_name bladnoch.elixirtech.com;
  ssl on;
  ssl_Certificate /opt/cert/elixirtech.pem;
  ssl_certificate_key /opt/cert/elixirtech.key;

  location / {
    proxy_pass http://localhost:8080/;
  }
}
```

In this case, the internal port is set to 8080.

5.2.2. High Availability Deployment

High Availability (HA) refers to systems that are durable and likely to operate continuously without failure for a long time.

If the system gets split, one side will have the majority of the machines, and that side will continue until the problem is resolved. Taking Ambience for example, the simplest HA setup will be two Ambience servers and a MongoDB server with a load balancer. A full HA setup will be two+ Ambience server and one MongoDB replica set with load balancers.

A load balancer (typically Nginx) is on a separate machine.

The diagram below shows a HA deployment.
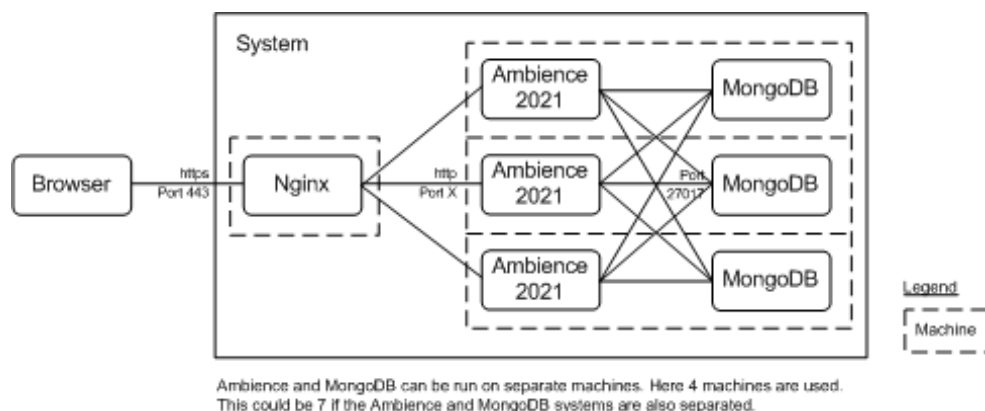
The Ambience and MongoDB can be run on separate machines. To avoid the single point of failure, another Nginx machine can be added and the DNA name server can be setup with an A record for each Nginx machine and a short TTL.

For cost saving, the three MongoDB servers can be replaced by two with one lightweight "Arbiter", which does not store data, but acts as the third machine in the case of a split.

Docker can be used to simplify deployment of these machines, and higher-level tools (such as, Docker Compose, Kubernetes, etc.) can be used to deploy an entire set of machine configurations at once. If the management of the deployment becomes tedious, then it might be worth investing that time to use these tools.

5.3. Specify Number of Threads (JVM)

The default minimum and maximum number of threads are set to four in Ambience.

You can override the default minimum and maximum number of threads using these values in the "run-server" script in the "/bin" folder:

```
-Dscala.concurrent.cntent.minThreads=X
-Dscala.concurrent.cntent.maxThreads=Y
```

where X and Y are the new minimum and maximum values respectively.

Note that the "threads" mentioned here are not hardware threads (which are fixed by the processor design, usually 1 core = 2 threads) but software threads, managed by the JVM.

The ideal values will depend on the load, i.e., number of clients, scheduled jobs, etc. Editing these values may help resolve certain blocking issues when mixing composites and reports (old synchronous code) with new asynchronous code.

5.4. Other Configurations

You can use the application configuration file (*application.conf*) to configure other settings, such as, upload file size, disabling ETL logs, etc.

Refer to [Application Config File Guide](#) for more information.