Notification

Table of contents

1. Setup E-Mail Notification

- To enable notification by e-mail, you have to download the [JavaMail](#) (http://java.sun.com/products/javamail/) and [JavaBeans Activation Framework](#) (http://java.sun.com/products/javabeans/jaf/downloads/index.html) libraries and place them in the `<lenya-webapp>/WEB-INF/lib` directory.

- Edit `cocoon.xconf`, look for the component with the role `org.apache.lenya.notification.Notifier`, change the class to `org.apache.lenya.notification.EmailNotifier` and add the `<smtp>` element as shown in the example below. The user and password attributes are optional.

```
<component logger="lenya.notification"
    role="org.apache.lenya.notification.Notifier"
    class="org.apache.lenya.notification.EmailNotifier">
  <smtp host="localhost" username="john" password="swordfish" />
</component>
```

- If you test the functionality with the default users lenya and alice, don't forget to configure their e-mail addresses.

- To check if the notification works, go to your [Inbox](#) (http://localhost:8888/default/authoring/index.html?lenya.usecase=notification.inbox) and send a message.

2. How to Send Notification Messages

There are two ways to send notification messages - either immediately, or by attaching a notification event to the current session. In the latter case, the notification is sent only if the transaction has been successfully completed.

To send notification messages, use the `Notifier` component which is provided by the `notification` module. A shortcut to the `Notifier` functionality is offered by the `NotificationUtil` utility class. Here's an example how to send a notification message immediately:

```
protected void notify(User sender, User recipient, document) {

    Identifiable[] recipients = { recipient };

    // compose message, using parameters for i18n

    String subject = "publish-notification";
    String[] subjectParams = {};

    String body = "document-was-published"
    String[] bodyParams = { document.getId() };
    Message message = new Message(subject, subjectParams, body, bodyParams,
                                  sender, recipients);

    // send message

    NotificationUtil.notify(this.manager, message);
}
```

The following example shows how to attach a notification event to the session, for instance inside a usecase handler:

```
NotificationEventDescriptor descriptor = new NotificationEventDescriptor(message);
RepositoryEvent event = RepositoryEventFactory
    .createEvent(this.manager, authoringDocument, getLogger(), descriptor);
getSession().enqueueEvent(event);
```

The message is sent to each particular recipient, translated using the recipient's locale.

The default implementation, `EmailNotifier`, sends an email to each recipient and adds the message to their inboxes. Future implementations could include a message list which is managed by the CMS and is presented to a user after she logs in.