Working with Documents

Table of contents

1. The Session

An `o.a.l.cms.repository.Session` is a temporary container for repository nodes which you want to work with. If you want to change or remove nodes - for instance in a usecase handler - you have to start a transaction. To avoid overriding or losing someone else's changes, you should lock any nodes which are potentially affected or read during your transaction.

A convenient way to get the session which is attached to the current request is provided by the `RepositoryUtil`:

```
Session session = RepositoryUtil.getSession(this.manager, request);
```

2. The Document Factory

The `o.a.l.cms.publication.DocumentFactory` is the main entry point to the content repository. It is tied to a session. You get the document factory which is attached to the current session this way:

```
DocumentFactory factory = DocumentUtil.getDocumentFactory(this.manager, request);
```

3. Browsing Content and Site Structure

From the document factory, you can access a publication:

```
String webappUrl = ServletHelper.getWebappUrl(request);
URLInformation info = new URLInformation(webappUrl);
Publication pub = factory.getPublication(info.getPublicationId());
```

The publication provides access to all areas (pun intended). An area object enables you to obtain documents by their UUID.

```
Area authoring = pub.getArea("authoring");
Document[] docs = authoring.getDocuments();
Document doc = authoring.getDocument(uuid, language);
```

If you want to obtain a document by its path in the site structure, get the site structure from the area:

```
SiteStructure site = authoring.getSite();
SiteNode node = site.getNode("/tutorial");
String[] languages = node.getLanguages();
Link link = node.getLink(language);
Document doc = link.getDocument();
```

You can also browse the document structure in a bottom-up way:

```
Document doc = ...;
doc.area().getPublication();
String area = doc.getLink().getNode().getStructure().getArea();
```

The Document class allows to access different language and area versions of the document:

```
if (doc.existsTranslation("en")) {
    englishVersion = doc.getTranslation("en");
}
if (doc.existsVersion("live", doc.getLanguage()) {
```

```
    addInfoMessage("Live version exists!");
    liveVersion = doc.getVersion("live", doc.getLanguage());
}
```