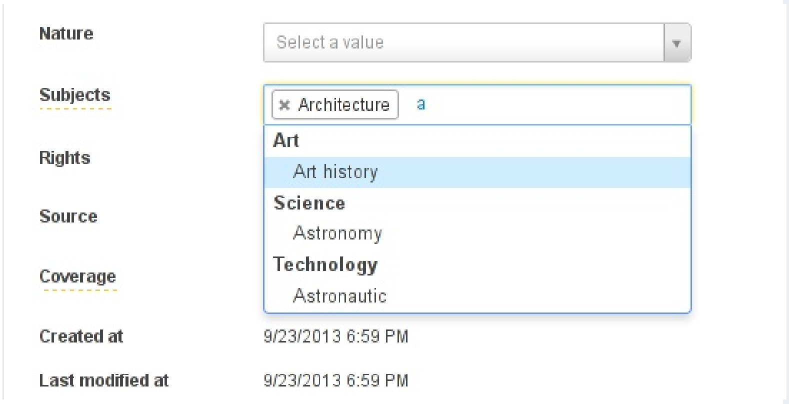
Nuxeo Platform 5.7.3 Release Notes

For Users

UI/UX Improvements

Select2 Integration

We have replaced the old selection/suggestion JSF widgets by select2 based widgets. Not only does it look better but it's also easier to customize. You can easily setup a user suggestion widget that filter search query with a group parameter.

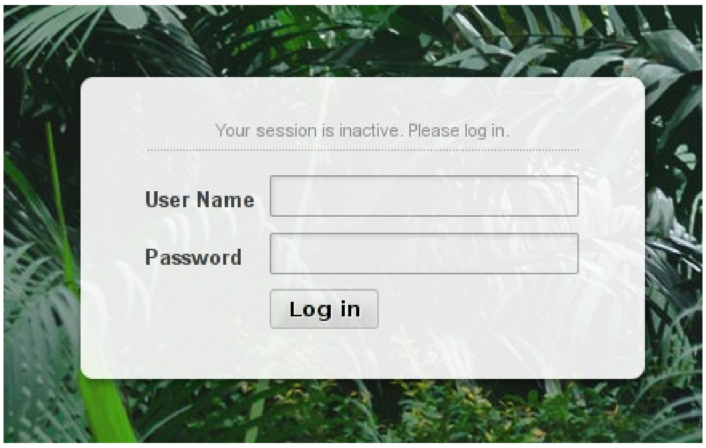


Nature	Select a value
Subjects	× Architecture a
Rights	
Source	
Coverage	
Created at	9/23/2013 6:59 PM
Last modified at	9/23/2013 6:59 PM

We also introduced new suggestion widgets on directories. They support n-level chained vocabularies out of the box.

SessionTimeOut Feedback

Users see a *'session expired'* message when their session expires.



Your session is inactive. Please log in.

User Name	
Password	

Form Submit Shield

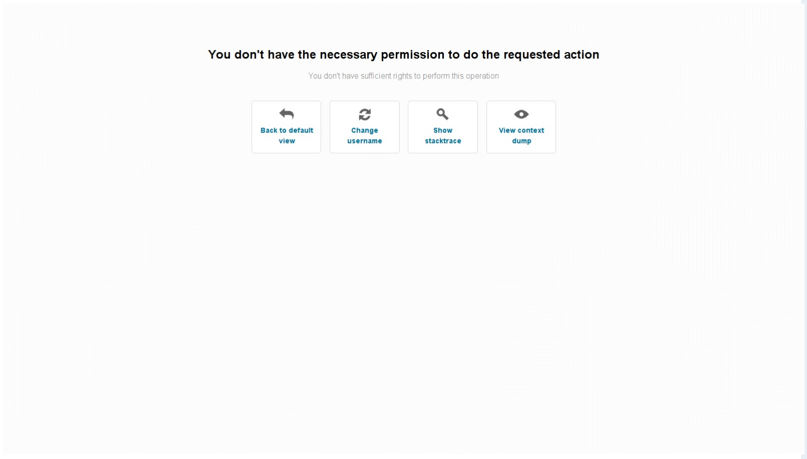
This will protect the heavy handed crazy clickers to have errors because they have clicked on the same button fifty times in two seconds. The user will receive a feedback message telling them the request is being processed. It's better for his mouse and the server. This is not activated by default, you need to set the *nuxeo.jsf.enableDoubleClickShield* property to true in *nuxeo.conf*.

Tag to Version

When a version of a document is created, the tags on the master document are copied to the new version.

Redesigned Error Page

We have redesigned the legacy error page to have something cleaner, with a user-understandable error message.



You don't have the necessary permission to do the requested action

You don't have sufficient rights to perform this operation

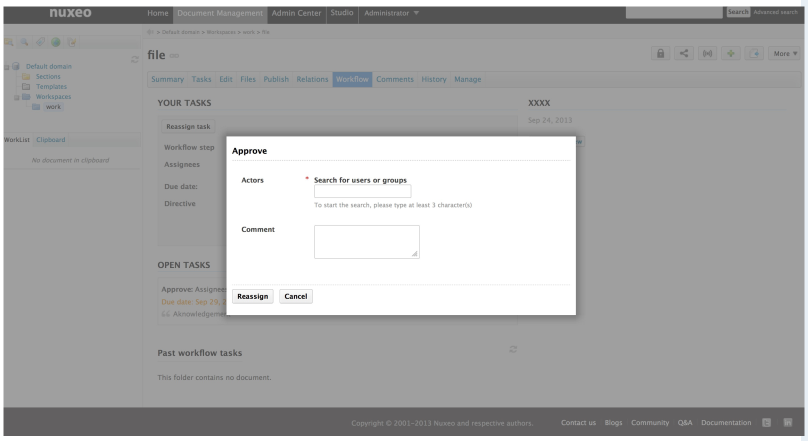
Workflow

New Workflow Tab

We have a new workflow tab that lets the user see all his assigned tasks at once and filter them.

Task Reassignement

Workflow tasks can now be reassigned to someone else.

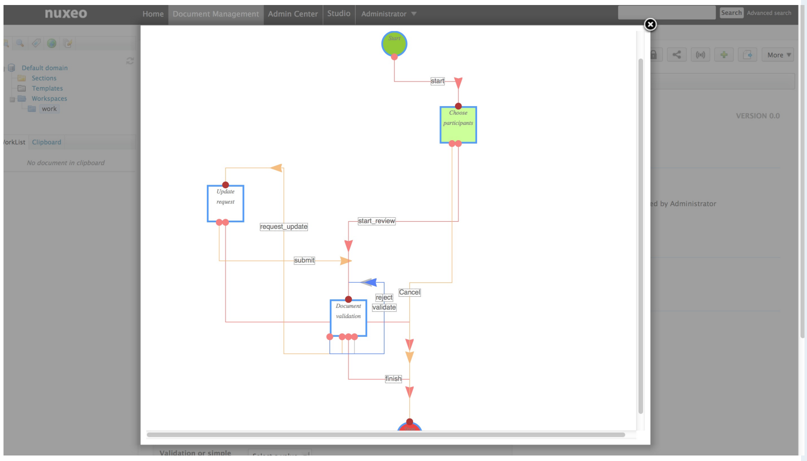


MultiTask Workflow Node

We have introduced a new option that creates a different task per assignee.

Workflow Graph Visualization

The workflow graph can be visualized as it has been created in Studio.



For Developers

API

New Endpoints

This is the major improvement of the 5.7.3 release. Nuxeo REST API is available on a Nuxeo Server at the following URL: http://localhost:8080/nuxeo/api/

Here's the list of endpoints available at the moment:

- Documents (/nuxeo/api/id/{docId}): to do CRUD on documents (including paginated search)
- Users (/nuxeo/api/user/{userId}): to do CRUD on users
- Groups (/nuxeo/api/group/{groupId}): to do CRUD on groups
- Directories (/nuxeo/api/directory/{directoryId}): to do CRUD on directories
- Automation (/nuxeo/api/automation/{Operation id}): to call a "command", i.e. an operation or operation chain deployed on the server

New Adapters

You can also use several adapters for ACLs, blob, business objects, document children, operation, page provider, search etc... An adapter is a URL segment that starts with "@" and that transforms the input resource so as to return another resource. For example, using *@blob* will return the file of a document (the one store on the property given by the next URL segment), and chaining it to *@op* will call an operation (that takes in input a blob):

```
/nuxeo/api/id/{docId}/@blob/file:content/@op/Blob.ToPDF
```

Pluggable Context

Because it is sometimes useful to optimize the number of requests you send to the server, we provide a mechanism to request more information on the answer, simply by specifying the context you want in the request header. You can activate contributor simply by adding the right header on the request. Those contributors can also be activated using action filter.

Automation

Trace

You can now enable tracing of Content Automation call. Operation chains and operations calls information are collected and logged during their execution by the Automation Trace feature.

Error management

Automation exception chain can be added to be executed when an error occurs during an Automation execution. After contributing your custom chains, you can declare your exception chains:

```
<extension point="chainException"
  target="org.nuxeo.ecm.core.operation.OperationServiceComponent">

    <catchChain id="catchChainA" onChainId="contributedchain">
      <run chainId="chainExceptionA" priority="0" rollBack="true" filterId="filte
      <run chainId="chainExceptionA" priority="0" rollBack="false" filterId="filt
      <run chainId="chainExceptionB" priority="10" rollBack="true" filterId="fil
    </catchChain>

    <catchChain id="catchChainB" onChainId="anothercontributedchain">
      <run chainId="chainExceptionA" rollBack="false"/>
    </catchChain>

</extension>
```

New Description Parameter for Operation

When you write a new operation, you can add a description to your declared parameters. This comes in handy for the API documentation and Nuxeo Studio.

UI/UX Framework

Action Context

Until now, action filters were using JEXL for resolution instead of the engine used by JSF, and Seam components could be resolved in this context, but using a hack, hence only actions *starting with the seam component name* could be resolved correctly, for instance #{clipboardActions.canCopy}. Others were not resolved correctly, for instance #{!clipboardActions.canCopy}, #{empty clipboardActions.selectedDocuments}, #{clipboardActions.canCopy and clipboardActions.canPaste}, etc...

Now the action context has been changed into an interface, and is responsible for evaluating expressions held by filters (instead of the filter itself) => by implementing this interface, *any kind of expression with any kind of variables* in context can be resolved using the filters/actions service API. Seam JSF EL are resolved in a native Seam/JSF context.

Action Widget Types

We've made a pluggable action types library. It means that now you can define your own action type and associated rendering. You can use it within new action widget types (like ToolBar actions, Form actions, Tabs actions etc...).

Runtime Service Seam Injection

You can now inject Nuxeo runtime services directly in a Seam bean. Where you use to call *Framework.getLocalService(SchemaManager.class)* in a method, you can simply inject the service.

```
    @In
    SchemaManager theSchemaManagerService;
```

Miscellaneous

Simple Computed Group

The idea is to simplify computed groups management by a simple contribution. Here are two examples. The first one will create a virtual group *grade_companyValue* where companyValue is the company value for the user. The second one will create a *creator_documentId* for each document where the user is the creator.

```
<extension point="computer" target="org.nuxeo.ecm.platform.computedgr
  <!-- userAttibute -->
  <userMetadataGroupComputer xpath="company" groupPattern="grade_
  <!-- documentMetadata -->
  <documentMetadataGroupComuter xpath="dc:title" whereClause="dc:cr
</extension>
```

Nuxeo Drive

Drive improvements are mainly bug fixes. We now handle transparently server-side version restoration. We have switched from PySide to PyQt for the GUI. We've also worked on the error code return by the server. The user will receive a 403 if credentials are wrong and a 401 if he don't have sufficient permission on a document.

ScanImporter Enhancement

We're still improving the scan importer. One of the latest enhancement is the support of MVEL in the XML mapping code. This will allow you to setup some business logic during import without writing cutom Java code. Here's an example:

```
<extension target="org.nuxeo.ecm.platform.importer.xml.parser.XMLImporte
  point="documentMapping">

  <docConfig tagName="html">
    <docType>Instruction</docType>
    <parent><![CDATA[ #{
      nodes = xml.selectNodes('//meta[@name=\'RCDirection\']/@content');
      String parent = nodes.get(0).getText();
      return Fn.mkdir(root, '/', parent ,'StructureFolder');
    }]]></parent>
    <name><![CDATA[ #{
      String valueFound = xml.selectNodes("//meta[@name='RCIdentifiant']/@
      String name = valueFound.replace(' ', '').replace('/', '-');
      return name;
    }]]></name>
    <postCreationAutomationChain>testBJA</postCreationAutomationChain
  </docConfig>

</extension>
```

With the following xml fragment:

```
<html>
  <head>
    <meta name="RCDirection" content="Dir1/Sec1.1" />
    <meta name="RCIdentifiant" content="DGAL/C98 - 8010" />
  </head>
...
```

Will be equivalent to the following code snippet:

```
// This computation is because the parent evaluation with the Fn.mkdir part
// This happends if the Dir1/Sec1.1 documents doesn't exist in root
// => see Fn.mkdir description above
String path = root.pathAsString;
DocumentModel doc = session.createDocumentModel(path, "Dir1", "Structu
doc = session.createDocument(doc);

path = doc.getPathAsString();
doc = session.createDocumentModel(path, "Sec1.1", "StructureFolder");
doc = session.createDocument(doc);

// Here is because the document creation activation
// Parent MVEL expression return the result of the mkdir
path = doc.getPathAsString();
// node selected into the xml is "DGAL/C98 - 8010"
// MVEL expression remove space and replace slash by minus
String name = "DGA-C98-8010";
doc = session.createDocumentModel(path, name, "Instruction");
```

NXQL

See the NXQL Documentation for the current state of NXQL.

ecm:isCheckedIn: 1 if the document is checked in and 0 if not (the opposite of DocumentModel.isCheckedOut()). This can only be compared to 1 or 0.

ecm:isVersion or ecm:isCheckedInVersion: 1 for versions and 0 for non-version (DocumentModel.isVersion()). This can only be compared to 1 or 0. (The name ecm:isVersion is available since Nuxeo 5.7.3)

ecm:versionVersionableId: the id of the versionable document of a version (the versionable document is the one from which the version was created).

ecm:isLatestVersion: 1 if this is the latest version of a document, 0 if not. This can only be compared to 1 or 0.

ecm:isLatestMajorVersion: 1 if this is the latest major version of a document, 0 if not. This can only be compared to 1 or 0.

For Administrators

Nuxeo.conf New Parameters

We've added Automation Tracing. Automation chains and operations calls information is collected during their execution by the Automation Trace feature. This Automation trace mode can be enabled through the nuxeo.conf file properties:

Parameter	Default Value	Description
org.nuxeo.automation.trace	false	Since Nuxeo 5.7.3, you can enable this mode if you'd like to display automation traces during runtime: • You'll benefit from exhaustive logs to debug all automation chain and/or operation execution. • The automation trace mode is disabled by default (not suitable for production). • It can be activated through JMX via org.nuxeo:TracerFactory MBean during runtime.
org.nuxeo.automation.trace.printable	*	Since Nuxeo 5.7.3, by default, all automation executions are 'printable' (appear in logs) when automation trace mode is on. • You can filter chain and/or operation execution trace printing by setting this property to chain name and/or operation separated by comma. • Comment this property to get all automation chains/operations back in printing (by default set to * (star)
nuxeo.jsf.enableDoubleClickShield	false (unset)	Enable a shield on forms to prevent users from submitting twice the same form (accidental double-click)