Nuxeo Server LTS 2017 Release Notes

Updated: **January 8, 2018** ⓘ ✎ [(https://github.com/nuxeo/doc.nuxeo.com-content/tree/master/src/nxdoc/nuxeo-server/nuxeo-server-release-notes.md)](https://github.com/nuxeo/doc.nuxeo.com-content/tree/master/src/nxdoc/nuxeo-server/nuxeo-server-release-notes.md)

This page relates to the release notes of Nuxeo Server and related addons for the 9.10 cycle, a.k.a LTS 2017 cycle. It will list the improvements and features that are successively shipped with the 9.1, 9.2, 9.3 and LTS 2017 releases. Evolutions are grouped by components. You can also find detailed JIRA release notes:

- [9.1 JIRA release notes (https://jira.nuxeo.com/secure/ReleaseNote.jspa?projectId=10011&version=17991)](https://jira.nuxeo.com/secure/ReleaseNote.jspa?projectId=10011&version=17991)
- [9.2 JIRA release notes (https://jira.nuxeo.com/secure/ReleaseNote.jspa?projectId=10011&version=18236)](https://jira.nuxeo.com/secure/ReleaseNote.jspa?projectId=10011&version=18236)
- [9.3 JIRA release notes (https://jira.nuxeo.com/secure/ReleaseNote.jspa?projectId=10011&version=18310)](https://jira.nuxeo.com/secure/ReleaseNote.jspa?projectId=10011&version=18310)
- [9.10 JIRA release notes (https://jira.nuxeo.com/secure/ReleaseNote.jspa?projectId=10011&version=18106)](https://jira.nuxeo.com/secure/ReleaseNote.jspa?projectId=10011&version=18106)

We also provide [instructions for upgrading (/nxdoc/upgrade-from-lts-2016-to-lts-2017/)](/nxdoc/upgrade-from-lts-2016-to-lts-2017/) to the latest release.

Nuxeo Server

Runtime

Runtime Components Lifecycle Refactored

The runtime behavior has been refactored so as to provide a cleaner lifecycle to its components. This (big) change is pretty transparent for now but will allow to provide new features to the runtime in the next releases.

➡ More on JIRA ticket [NXP-19326 (https://jira.nuxeo.com/browse/NXP-19326)](https://jira.nuxeo.com/browse/NXP-19326).

Change in Runtime Reload Strategy

Runtime reload strategy has been set by default to `standby` and can be changed with the property `org.nuxeo.runtime.reload_strategy`. Standby mode will:

- Stop all components
- Apply the stash (components to remove/add before restarting)
- Start all components

`restart` will:

- Stop all components
- Deactivate all components
- Apply the stash
- Activate all components
- Start all components

➡ More on JIRA ticket NXP-22546 (https://jira.nuxeo.com/browse/NXP-22546).

Update Mechanism When Reloading Components

Hot-reload has been optimized so as to limit the number of components that are reloaded, also making sure that they are loaded only once during the process.

➡ More on JIRA ticket NXP-23166 (https://jira.nuxeo.com/browse/NXP-23166).

Graceful Shutdown in Tomcat

Runtime behavior has been cleaned up: when shutting down Tomcat it is now possible to make sure that all asynchronous work has been either processed or task has been persisted.

➡ More on JIRA ticket NXP-21969 (https://jira.nuxeo.com/browse/NXP-21969).

Backing Service

We added the infrastructure for plugging some initial checks on external services availability *before* starting the runtime. The first use case is to add a check for MongoDB or PostgreSQL availability, when the MongoDB or PostgreSQL template is used. In case it is not available, the nuxeo-launcher is interrupted.

➡ More on JIRA ticket NXP-21532 (https://jira.nuxeo.com/browse/NXP-21532).

Better Error Message on Invalid Class in Extension Point Contribution

When a contribution to the runtime expects a class name, and the effective contribution does not refer to a class, runtime exception has been made clearer and there is now a warning at startup. Ex: "Failed to load contributions for component service:org.nuxeo.ecm.core.cache.config.bad".

➡ More on JIRA ticket NXP-22797 (https://jira.nuxeo.com/browse/NXP-22797).

Test @PartialDeploy Annotation

The new Test `@PartialDeploy` annotation allows you to select exactly which contributions you want to deploy from a component.

```
@PartialDeploy(bundle = "studio.extensions.YOUR_PROJECT_NAME", extensions
= { TargetExtensions.Automation.class })
```

It makes it easier to test Nuxeo Studio features like Automation Scripting or event handlers.

Core Repository

Centralized Automated Versioning Policies

A new extension point has been added to define automated versioning policies. It allows to declare behaviors such as "Version documents each time the current contributor is different than the last one on the document". It also allows to control whether you want to version before or after the modification. All platform code that was triggering specifically hardcoded behaviors has been refactored so that the central policy is applied: Drive, File Manager (drag and drop behaviors), etc.). Only CMIS hasn't been impacted for now due to some difficulties with requirements regarding document checked in or checked out.

➡ More on JIRA ticket NXP-21588 (https://jira.nuxeo.com/browse/NXP-21588), the versioning developer documentation (/nxdoc/versioning/) or the user documentation (/userdoc/).

More Permissions Returned in the Permissions Enricher

All visible permissions and more core permissions are included by the Permissions enricher (http://community.nuxeo.com/api/nuxeo/latest/javadoc/org/nuxeo/ecm/core/io/marshallers /json/enrichers/BasePermissionsJsonEnricher.html). Permission enricher can be used to know what permissions the authenticated user has on the document.

➡ More on JIRA ticket NXP-21408 (https://jira.nuxeo.com/browse/NXP-21408) and the Content Enricher documentation (/nxdoc/content-enrichers/).

Orphan Versions Removal

In some cases, versions of some documents where not deleted even when the document itself was removed. The solution chosen to solve this issue has been to add a scheduled cleanup mechanism. The cleanup mechanism does periodic commits, every 1000 documents by default. This value can be changed using the configuration property `org.nuxeo.orphanVersionsCleanup.commitSize` (http://explorer.nuxeo.com/nuxeo/site/distribution/Nuxeo%20Platform-9.1/viewContribution /org.nuxeo.ecm.core.orphanVersionsCleanup--configuration).

➡ More on JIRA ticket NXP-14187 (https://jira.nuxeo.com/browse/NXP-14187).

Document Initialization (Empty Document) REST Endpoint

A new web adapter allows returning an empty document model so as to get default values (and listeners impact on `emptyDocumentCreated` event) client side. The adapter is named `@emptyWithDefault`. Here are some examples on how to use it:

- `/@emptyWithDefault?emptyDocType=File&emptyDocName=toto`
- `path/{docPath}/@emptyWithDefault`
- `id/{docId}/@emptyWithDefault?emptyDocType=File&emptyDocName=toto`

➡ More on JIRA ticket NXP-23156 (https://jira.nuxeo.com/browse/NXP-23156).

Repository Configuration Endpoint

The following configuration endpoints were missing from the API doc:

- `/config/facets`
- `/config/types`
- `/config/schemas`

They are now available from your Nuxeo instance REST API explorer at `http://NUXEO_SERVER/nuxeo/api/v1/doc`.

➡ More on JIRA ticket NXP-22885 (https://jira.nuxeo.com/browse/NXP-22885).

Compatible with CloudFront for Caching Binaries

When using S3 it is possible to redirect signed CloudFront URLs instead of directly S3 ones, so as to benefit from AWS world wild content caching service.

➡ More on JIRA ticket NXP-20219 (https://jira.nuxeo.com/browse/NXP-20219).

Leverage Blob's XPath in Dispatching Rules

It is now possible to use the XPath of the binary that is being stored to determine where the binary should be stored. This typically allows to store thumbnails in a different backend than the main file. The DefaultBlobDispatcher can now match blob XPath using the syntax `blob:xpath=my/xpath` and a new glob operator `~` is available to allow matching things like `blob:xpath~files/*/file`.

➡ More on JIRA ticket NXP-21891 (https://jira.nuxeo.com/browse/NXP-21891).

Change Token for Optimistic Locking

New API:

- `DocumentModel.getChangeToken()`

New behaviors:

- Calling `doc.putContextData(CoreSession.CHANGE_TOKEN, token)` then `CoreSession.saveDocument(doc)` will check the change token and raise `ConcurrentUpdateException` in case of mismatch.
- Calling `doc.putContextData(CoreSession.USER_CHANGE, Boolean.TRUE)` then `CoreSession.saveDocument(doc)` will flag the save as a "user change".

The change token returned to the user-level API `DocumentModel.getChangeToken()` is now a mix of a system-change token (updated at every change) and a user-change token (updated at every user change).

- A "user change" is a change initiated by a user-level operation, which is defined as a document save that passes a previous change token using `doc.putContextData(CoreSession.CHANGE_TOKEN, token)`, or that marks the save as an explicit user change using `doc.putContextData(CoreSession.USER_CHANGE, Boolean.TRUE)`.
- The system change token is such that every modification to a document (whether done by the system or by a user) now increments a system value called the "system change token".

➜ More on JIRA tickets [NXP-19435 (https://jira.nuxeo.com/browse/NXP-19435)](https://jira.nuxeo.com/browse/NXP-19435), [NXP-22019 (https://jira.nuxeo.com/browse/NXP-22019)](https://jira.nuxeo.com/browse/NXP-22019) and [NXP-22259 (https://jira.nuxeo.com/browse/NXP-22259)](https://jira.nuxeo.com/browse/NXP-22259).

Document.Update Operation: Improved Change Token Detection

The `Document.Update` operation has been updated so that it is possible to use the change token mechanism and ignore system modifications, that were taken into account with previous releases.

➜ More on JIRA ticket [NXP-22751 (https://jira.nuxeo.com/browse/NXP-22751)](https://jira.nuxeo.com/browse/NXP-22751).

Ability to Set Change Token from REST API

When sending an update to a document with the REST API, the JSON can now include a `changeToken` field (as a toplevel field), with the same value that was retrieved previously when reading the document. A 409 will then be returned if the update is in conflict server-side.

➜ More on JIRA ticket [NXP-22254 (https://jira.nuxeo.com/browse/NXP-22254)](https://jira.nuxeo.com/browse/NXP-22254).

Retention Flag

A new method has been added to the CoreSession so as to set a flag `ecm:isRetentionActive` on a document in order to specify it is in retention:

```
session.setRetentionActive(doc.getRef(), true);
```
.

An event `retentionActiveChanged` is sent and an entry is logged in the audit trail when the method is called.

When a document is in retention, it can't be deleted using the delete API. The retention flag needs to be set to false prior to deletion. This can only be done by users with WriteSecurity permission.

➡ More on JIRA ticket [NXP-23165 (https://jira.nuxeo.com/browse/NXP-23165)](https://jira.nuxeo.com/browse/NXP-23165).

Allow for the Document to Not Be Saved in the Add/Remove Facet Operations

A new parameter has been added: `save`. When `true`, the document is saved, else it is not (other operations in the chain should then save it). Note: Default value is `true`, the document is saved by default to maintain compatibility.

➡ More on JIRA ticket [NXP-22912 (https://jira.nuxeo.com/browse/NXP-22912)](https://jira.nuxeo.com/browse/NXP-22912).

Disable Delta Computation

By adding the new option `org.nuxeo.core.delta.disabled=true` in the `nuxeo.conf` file, the delta calculation used by the [Nuxeo Quota addon (/nxdoc/nuxeo-quota/)](/nxdoc/nuxeo-quota/) can be skipped. This can improve the performance of mass import.

➡ More on JIRA ticket [NXP-20892 (https://jira.nuxeo.com/browse/NXP-20892)](https://jira.nuxeo.com/browse/NXP-20892).

API for Permission Purge

An operation `PermissionsPurge` has been added for triggering a permission purge for a given principal. This allows you to do an integration with your identity management system so as to remove any permission to a user that would be leaving your organization in the same time you would disable his credentials.

➡ More on JIRA ticket [NXP-20844 (https://jira.nuxeo.com/browse/NXP-20844)](https://jira.nuxeo.com/browse/NXP-20844).

Date Fields Export with Millisecond Precision

The XML Core IO format and the JSON serialization now include milliseconds information for all date properties.

➡ More on JIRA ticket [NXP-21607 (https://jira.nuxeo.com/browse/NXP-21607)](https://jira.nuxeo.com/browse/NXP-21607).

User Existence Is Checked on Document.AddPermission Operation

A check on the existence of users an groups referenced with this operation is performed before doing the assignment. An exception is raised otherwise.

➡ More on JIRA ticket [NXP-21559 (https://jira.nuxeo.com/browse/NXP-21559)](https://jira.nuxeo.com/browse/NXP-21559).

Atomic getOrCreateDocument Method

A new method is available in CoreSession: `getOrCreateDocument(DocumentModel)` . This method can be used to prevent different threads from trying to create the same document.

➜ More on JIRA ticket NXP-19009 (https://jira.nuxeo.com/browse/NXP-19009).

Attribute "perDocumentQuery" of Facet Definition Can Be Overridden

It is now possible to change the value of the attribute `perDocumentQuery` of a facet that has been contributed in the built-in distribution.

➜ More on JIRA ticket NXP-22603 (https://jira.nuxeo.com/browse/NXP-22603).

New Document Properties Deprecation

A new extension point `deprecation` (http://explorer.nuxeo.com/nuxeo/site/distribution/latest /viewExtensionPoint/org.nuxeo.ecm.core.schema.TypeService--deprecation) has been added to the Schema service to configure deprecated fields and their optional fall-back. A warning is logged when a deprecated field is used in the application, and if the fall-back is configured, values are set and gotten from that fall-back field.

```
<property schema="common" name="size" deprecated="true" />
<property schema="file" name="filename" fallback="content/name" />
```

➜ More on JIRA ticket NXP-21456 (https://jira.nuxeo.com/browse/NXP-21456).

Distributed Cache

Invalidation cache has been rewritten to a better distributed architecture that improves performances.

➜ More on JIRA ticket NXP-22786 (https://jira.nuxeo.com/browse/NXP-22786).

Core Storage

SSL Support for MarkLogic

When using the MarkLogic Connector for Nuxeo (https://connect.nuxeo.com/nuxeo/site/marketplace /package/nuxeo-marklogic-connector), connection is now secured with SSL.

➜ More on JIRA ticket NXP-21818 (https://jira.nuxeo.com/browse/NXP-21818).

Improved Performance on Scrolling API with MarkLogic

Scroll API (that allows to fetch many documents in a row) has been re-implemented on Marklogic connector (https://connect.nuxeo.com/nuxeo/site/marketplace/package/nuxeo-marklogic-connector).

Elastiscsearch re-indexation rate has been increased from 642 documents/second to 2,418 documents/second, i.e + 375%.

➡ More on JIRA ticket NXP-21824 (https://jira.nuxeo.com/browse/NXP-21824).

New Oracle JDBC Driver (ojdbc7) Compatibility

The new Oracle JDBC Driver OJBC7 can be used with Oracle 12c.

➡ More on JIRA ticket NXP-19373 (https://jira.nuxeo.com/browse/NXP-19373).

Optimized Number of Read When Writing

The number of read requests required to create documents has been reduced to the strictly required using cache. As a consequence CPU is no longer the limiting factor of bulk-intensive import scenari.

➡ More on JIRA ticket NXP-20595 (https://jira.nuxeo.com/browse/NXP-20595).

Database-Level Integrity Constraints for PostgreSQL

When using PostgreSQL, stricter database-level checks are now enabled to prevent rare document name collisions due to concurrent writes. Behavior can be controlled in the repository configuration file using

```
<childNameUniqueConstraintEnabled>true</childNameUniqueConstraintEnabled>
```

➡ More on JIRA ticket NXP-22421 (https://jira.nuxeo.com/browse/NXP-22421).

Database-Level Integrity Constraints for PostgreSQL BIS

When using Postgres, a UNIQUE INDEX has been added on several tables preventing from having the same entry twice: the list of contributors, as well as the hierarchy tables.

➡ More on JIRA ticket NXP-23198 (https://jira.nuxeo.com/browse/NXP-23198).

Exception When Using S3 storage With an Empty AWS_ACCESS_KEY_ID

A more meaningful error is thrown when the platform doesn't find `AWS_ACCESS_KEY_ID` when starting.

➡ More on JIRA ticket NXP-22967 (https://jira.nuxeo.com/browse/NXP-22967).

Core Events

A new PubSubService is available, providing cross-instance publish/subscribe methods:

- `publish(topic, message)` sends a message to the given topic
- `registerSubscriber(topic, subscriber)` registers a subscriber

➡ More on JIRA ticket NXP-21800 (https://jira.nuxeo.com/browse/NXP-21800).

Directory

administrators Group Members Have Access to Protected Directories

Directories access protected by giving `Read` permission to group "Nobody" are now accessible by members of the group "administrators".

➡ More on JIRA ticket NXP-22609 (https://jira.nuxeo.com/browse/NXP-22609).

New MongoDB Directory Implementation

Directory interface now has a MongoDB implementation included in the default distribution.

➡ More on JIRA ticket NXP-21582 (https://jira.nuxeo.com/browse/NXP-21582).

Simple Property to Switch Back to SQL Directories when MongoDB is Used

A property has been added to keep directories in SQL when using MongoDB `nuxeo.mongodb.directories.enabled` (that must be set to true for keeping directories on SQL).

➡ More on JIRA ticket NXP-23190 (https://jira.nuxeo.com/browse/NXP-23190).

Default Caching Configuration for Directories

If a cache name is not defined in the directory description, a cache named `directoryname-cache` will be created with the values of the default-cache.

➡ More on JIRA ticket NXP-17982 (https://jira.nuxeo.com/browse/NXP-17982).

Generic Directories References

A new descriptor for references in directories has been added (their description used to be specific to the SQL implementation of directories). Example:

```
<references>
        <reference field="groups" directory="groupDirectory"
                   name="user2group" source="userId"
                   target="groupId" dataFile="user2group.csv"/>
</references>
```

This reference is generic and can be used with SQL and MongoDB implementations.

➜ More on JIRA ticket NXP-22411 (https://jira.nuxeo.com/browse/NXP-22411).

Nuxeo Streams

An asynchronous infrastructure to process distributed work chains has been written and named "nuxeo-stream". It creates an abstraction on top of Kafka and Chronicle Queues (for the standalone implementation).

It provides two patterns:

- a batched producer/consumer pattern dedicated to import process
- a Stream Computation pattern (https://github.com/nuxeo/nuxeo-mqueues/tree/master/nuxeo-mqueues-core#pattern-4-stream-and-computations) (taken from Concord Mesos (http://concord.io/docs/guides/concepts.html), Google MillWheel or Kafka Processor (https://cwiki.apache.org/confluence/display/KAFKA/KIP-28+-+Add+a+processor+client) to chain tasks

nuxeo-stream is now used for several use cases in the platform, either by default or optionally: Audit (by default), Workers (optionally), Import. More usage will be integrated in the near future.

➜ More on JIRA ticket NXP-21993 (https://jira.nuxeo.com/browse/NXP-21993).

WorkManager

Implementation On Top of nuxeo-stream

The WorkManager now has an implementation on nuxeo-stream that provides better resilience and distribution capability.

➜ More on JIRA ticket NXP-22500 (https://jira.nuxeo.com/browse/NXP-22500).

New Metrics

Some metrics of the WorkManager added during this LTS 2017 cycle have been renamed. Overall we now have:

- `nuxeo.works.total.default.canceled` : The number of cancelled work persisted in Redis in cluster mode
- `nuxeo.works.total.default.completed` : The number of completed work persisted in Redis in cluster mode
- `nuxeo.works.total.default.running` : The current number of work running
- `nuxeo.works.total.default.scheduled` : The current number of work scheduled
- `nuxeo.works.default.completed` : The total number of completed work since this instance is up
- `nuxeo.works.default.running` : The current number of running work in this instance

- `nuxeo.works.default.scheduled` : The total number of work scheduled by this instance
- `nuxeo.works.default.total` : A timer that measures work execution in this pool

➞ More on JIRA ticket [NXP-22996 (https://jira.nuxeo.com/browse/NXP-22996)](https://jira.nuxeo.com/browse/NXP-22996).

Audit

MongoDB Audit Backend

A MongoDB backend has been implemented with the purpose of being able to install the Nuxeo Platform without requiring an additional relation database. Since 9.3, the implementation is available by default and doesn't require to install any addon.

➞ More on JIRA tickets [NXP-21500 (https://jira.nuxeo.com/browse/NXP-21500)](https://jira.nuxeo.com/browse/NXP-21500) and [NXP-22247 (https://jira.nuxeo.com/browse/NXP-22247)](https://jira.nuxeo.com/browse/NXP-22247).

Audit Writer Based on nuxeo-stream

A new audit synchronous listener and writer based on nuxeo-stream is activated by default. This provides a more reliable and performant solution than the previous audit bulk writer. The old implementation can still be used with the following option: `nuxeo.stream.audit.enabled = false` .

➞ More on JIRA ticket [NXP-22109 (https://jira.nuxeo.com/browse/NXP-22109)](https://jira.nuxeo.com/browse/NXP-22109).

New Audit APIs getLatestLogId, getLogEntriesAfter and queryLogs

Three new methods have been added to the Audit service: `getLatestLogId` , `getLogEntriesAfter` and `queryLogs`

➞ More on JIRA ticket [NXP-21661 (https://jira.nuxeo.com/browse/NXP-21661)](https://jira.nuxeo.com/browse/NXP-21661).

Audit Storage SPI

A new interface `AuditStorage` has been added to define a storage back-end abstraction, so as to be able to store audit data in several places easily. A "storage" doesn't implement all the methods of the Audit service, but enough for storing audit entries and scroll them.

➞ More on JIRA ticket [NXP-23291 (https://jira.nuxeo.com/browse/NXP-23291)](https://jira.nuxeo.com/browse/NXP-23291).

Restore From Storage

- A new extension point is available to register an audit storage:

```
<extension
target="org.nuxeo.ecm.platform.audit.service.NXAuditEventsService"
point="storage">
<storage id="defaultAuditStorage"
class="org.nuxeo.ecm.platform.audit.service.DefaultAuditBackend"/>
</extension>
```

- A new automation operation `Audit.Restore` is available to be able to restore an audit backend from an audit storage.

➜ More on JIRA ticket NXP-23292 (https://jira.nuxeo.com/browse/NXP-23292).

CMIS

Relax Mode

Some constraints of CMIS can be bypassed. To do so use the runtime property `nuxeo.dontFollowCmisSpec=true`. This way, for instance, multiple "contains" can be used in the CMISQL query while the standard forbids it.

➜ More on JIRA ticket NXP-19858 (https://jira.nuxeo.com/browse/NXP-19858).

Change Token

Change Token specified (http://docs.oasis-open.org/cmis/CMIS/v1.1/cs01/CMIS-v1.1-cs01.html#x1-1610003) in CMIS 1.1 as optional has been implemented.

➜ More on JIRA ticket NXP-20665 (https://jira.nuxeo.com/browse/NXP-20665).

Proxies Are Visible by Default

`org.nuxeo.cmis.proxies=true` is now the default.

➜ More on JIRA ticket NXP-21828 (https://jira.nuxeo.com/browse/NXP-21828).

maxItems Parameter in CMIS Query Is Now Supported

When using the ES implementation of CMIS bridge, using the parameter `maxItems` allows you to get a certain amount of documents without loading the entire result set at once. Implementation over Core in case of DBS has also been improved consequently.

➜ More on JIRA ticket NXP-21877 (https://jira.nuxeo.com/browse/NXP-21877).

Query

Search with Highlighted Results

It is now possible to configure a page provider so as to get highlighted search result, in order to display the paragraph wrapping a given full-text searched term, in the search result. The Search endpoint (/nxdoc/search-endpoints/) and PageProvider operation have been updated accordingly. Checkout the polymer sample (https://github.com/nuxeo-sandbox/nuxeo-highlight-showcase) that showcases this new capability of the API to know more about it. Since Nuxeo Platform 9.2, this is used by default in Nuxeo Web UI.

➜ More on JIRA ticket NXP-20507 (https://jira.nuxeo.com/browse/NXP-20507).

Search Rest Endpoint Exposes Page Provider Offset

The `search` endpoint now exposes an "offset" property which is used by the underlying page provider to return results from this offset. If set, the `currentPageIndex` property is ignored.

➜ More on JIRA ticket NXP-22775 (https://jira.nuxeo.com/browse/NXP-22775).

No Default Query Language for Search Endpoint Anymore

Query language parameter in `search` endpoint no more has a default.

➜ More on JIRA ticket NXP-22754 (https://jira.nuxeo.com/browse/NXP-22754).

Max Results in the Page Provider Response

`resultsCount` property has been added to the page provider response so as to let the client know what are the server side limits of search result computation. This value is configured server-side with the property `maxResults` .

➜ More on JIRA ticket NXP-23202 (https://jira.nuxeo.com/browse/NXP-23202).

Conversion

PDF Export of a Note

It is now possible to export a Note with its embedded images as a PDF.

➜ More on JIRA ticket NXP-16831 (https://jira.nuxeo.com/browse/NXP-16831).

FFmpeg Embedded AAC Encoder Instead of libfaac

FFmpeg embedded AAC encoder is now used instead of libfaac, simplifying the set up phase.

➜ More on JIRA ticket NXP-22777 (https://jira.nuxeo.com/browse/NXP-22777).

Rendition

defaultRendition Extension Point to Rendition Service

In the same way we are able to easily customize the download service's permission extension point (described in File Download Security Policies (/nxdoc/file-download-security-policies/)), a new extension point allows to dynamically evaluate a default rendition to be generated for a given document. This evaluation is based on the JVM Nashorn JavaScript engine and looks like:

```xml
<extension
target="org.nuxeo.ecm.platform.rendition.service.RenditionService"
point="defaultRendition">
<defaultRendition name="nxDefaultRendition">
<script language="JavaScript">
function run() {
if (Reason == 'download') {
if (Document.getType() == "File") {
return 'mainBlob';
} else if (Document.getType() == 'Folder') {
return '';
} else {
return 'xmlExport';
}
} else {
return '';
}
}
</script>
</defaultRendition>
</extension>
```

This default rendition is now used for bulk downloading documents. This way, users can easily customize how a document type can be rendered within the context of a bulk download. For instance, within a zip resulting from a bulk download, documents without any blob attached can be XML-rendered, Picture documents will have their main blob rendered, Note will be rendered as PDF, etc.

➙ More on JIRA ticket NXP-23258 (https://jira.nuxeo.com/browse/NXP-23258).

Operation to Bulk Download the Main Rendition

The `Blob.BulkDownload` operation allows to bulk download all the main renditions provided by each of the documents set in the input of that operation. It returns an asynchronous blob information.

➙ More on JIRA ticket NXP-23063 (https://jira.nuxeo.com/browse/NXP-23063).

Elasticsearch

Make use of Elasticsearch 5.x instead of 2.X

Elasticsearch's Nuxeo Platform dependency is now on version 5.6 of the search engine. This upgrade brings security and performance improvements in addition to additional query capabilities. More specifically, the connector now uses the HTTP(s) interface of Elasticsearch (the Java one is still available).

➡ More on JIRA ticket NXP-22812 (https://jira.nuxeo.com/browse/NXP-22812).

ES Rest Client Supports TLS and BasicAuth

The `ElasticSearchClient` extension point can be configured to enable Basic Auth:

```
<elasticSearchClient
class="org.nuxeo.elasticsearch.clientESRestClientFactory">
<option name="addressList">localhost:9300</option>
<option name="username">jdoe</option>
<option name="password">secret</option>
</elasticSearchClient>
```

and SSL using `keystore.path` and `keystore.password options` .

Socket and connection timeout are also configurable with `socket.timeout.ms` and `connection.timeout.ms` .

➡ More on JIRA ticket NXP-23060 (https://jira.nuxeo.com/browse/NXP-23060).

Shield Support

Elasticsearch Shield support has been added since Nuxeo Platform 9.1. It is possible to configure authentication credentials in nuxeo.conf (/nxdoc/configuration-parameters-index-nuxeoconf/) (`elasticsearch.shield.enabled` , `elasticsearch.shield.username` , `elasticsearch.shield.password`). It also includes support of SSL encryption of the traffic since Nuxeo Platform 9.2.

Warning: in 9.3 this feature has been removed as Elasticsearch implementation was migrated to 5.X. A compatibility with X-Pack will be added later.

➡ More on JIRA tickets NXP-21208 (https://jira.nuxeo.com/browse/NXP-21208) and NXP-22042 (https://jira.nuxeo.com/browse/NXP-22042), and the Elasticsearch Shield documentation (/nxdoc /elasticsearch-setup/#configuring-access-to-the-cluster-through-elasticsearch-shield-plugin) for more information.

Re-indexing Without Downtime

Re-indexing the repository can be done with no downtime as new parameters have been added to the index descriptor:

- `manageAlias=true` : Nuxeo will manages 2 aliases: one for searching using the name of the contrib (default to `nuxeo`), one for writing with a "-write" suffix (`nuxeo-write`). Both aliases will point to the same index (`nuxeo-0000`). When reindexing the repository a new index is created (`nuxeo-0001`) and the write alias is updated to use it. The search alias stays on the previous index (`nuxeo-0000`). Once indexing is terminated the search alias is updated to point to the new index (`nuxeo-0001`). It is up to the administrator to clean old non-used indexes (keep the 2 last created for instance)

- `writeAlias` : When specified the write is done on this alias, and nuxeo will not manage any aliases. The use case is for managing time-based index for audit: One can create a new index every month the search alias can point to multiple indexes.

➡ More on JIRA ticket NXP-191902 (https://jira.nuxeo.com/browse/NXP-191902).

Amazon Elasticsearch Service is Supported

Nuxeo Platform has been made compatible with Amazon Elasticsearch Service to replace the Elasticsearch cluster in the default architecture.

➡ More on JIRA ticket NXP-23134 (https://jira.nuxeo.com/browse/NXP-23134).

Elasticsearch PageProvider Limits Navigation to 10k Documents (Configurable)

The page provider navigation is now limited to the first 10k documents. This is the default maximum result window defined in Elasticsearch.

When there are more than 10k results there is no more button to go to the last page or to go to a next page above this range. The limit of 10k can be changed using the configuration service by contributing the property `org.nuxeo.elasticsearch.provider.maxResultWindow` to the Configuration service (http://explorer.nuxeo.com/nuxeo/site/distribution/latest/viewExtensionPoint /org.nuxeo.runtime.ConfigurationService--configuration#contribute).

➡ More on JIRA ticket NXP-22063 (https://jira.nuxeo.com/browse/NXP-22063).

ecm:acl Removed from the Fulltext in the Default Elasticsearch Configuration

ACLs are no more indexed on the fulltext index.

➡ More on JIRA ticket NXP-22788 (https://jira.nuxeo.com/browse/NXP-22788).

Use Elasticsearch Tools with Embedded Deployment of ES

It is now possible to use Elasticsearch tools to inspect (like elasticsearch-head) and query (like mirage) on Nuxeo index when Elasticsearch is embedded.

➜ More on JIRA ticket NXP-21455 (https://jira.nuxeo.com/browse/NXP-21455).

Tag Service

New Implementation Compatible With MongoDB

A new implementation for the Tag service has been made available by default. This implementation is available on MongoDB and MarkLogic (while the previous implementation only worked for VCS). A property allows switching the implementation to the former one if necessary: `nuxeo.faceted.tag.service.enabled` . New tag implementation stores values as properties of the document on a dedicated facet. As a consequence, the document type definition shall include that facet to benefit from Tag service support on a given document. A migration procedure has been written to easily handle the migration from legacy implementation to the new one.

➜ More on JIRA ticket NXP-21651 (https://jira.nuxeo.com/browse/NXP-21651).

Migration Procedure for Tag Service

A migrator has been added for tag service migration. When upgrading from a previous Nuxeo version, if you want to keep (and later migrate) old tags, add the contribution:

```
<require>org.nuxeo.ecm.platform.tag.service.migrator</require>
<extension target="org.nuxeo.runtime.migration.MigrationService"
point="configuration">
<migration id="tag-storage">
<defaultState>relations</defaultState>
</migration>
</extension>
```

This was the occasion to implement a whole new migration model that can be reused when implementing major evolutions.

➜ More on JIRA ticket NXP-23091 (https://jira.nuxeo.com/browse/NXP-23091).

User Manager

New Operations

New automation operations are available:

- `User.CreateOrUpdate`
- `Group.CreateOrUpdate`

Both operations have a "mode" parameter that can be set to "create" or "update" to force the

operation to do only creates or only updates. Both operations can receive direct parameters or use a "properties" map to receive them.

The `Services.Query` operation has been renamed `User.Query`.

➡ More on JIRA tickets NXP-21627 (https://jira.nuxeo.com/browse/NXP-21627) and NXP-21962 (https://jira.nuxeo.com/browse/NXP-21962).

FileManager

A new parameter allows to avoid overriding an existing folder.

➡ More on JIRA ticket NXP-21853 (https://jira.nuxeo.com/browse/NXP-21853).

Redis

Redis 4.0 supported

Both Redis 3.2 and 4.0 are now supported.

➡ More on JIRA ticket NXP-21770 (https://jira.nuxeo.com/browse/NXP-21770).

Key Value Store

A New Service Implemented On Redis

A Key/Value store has been added to the Core API and implemented in the Redis addon. From the KeyValueService one can acquire a named KeyValueStore, whose API is currently:

- `put(key, value)`
- `get(key)`
- `compareAndSet(key, expected, value)`

➡ More on JIRA ticket NXP-21870 (https://jira.nuxeo.com/browse/NXP-21870).

TTL Management On Key/Value store

TTL support has been added to the KeyValueStore with two additional methods to set it:

- `put(key,value,ttl)`
- `setTTL(key,ttl)`

➡ More on JIRA ticket NXP-23029 (https://jira.nuxeo.com/browse/NXP-23029).

MongoDB Implementation

The Nuxeo Key / Value store now has a MongoDB-based implementation. This removes a dependency toward Redis, with the goal of not needing Redis when all dependencies will have been removed (there

will remain the pub/sub implementation at LTS 2017 release time).

➡ More on JIRA ticket NXP-23030 (https://jira.nuxeo.com/browse/NXP-23030).

OAuth

Nuxeo Platform as OAuth Provider

There was a first implementation of the OAuth flow to use Nuxeo as an OAuth provider. It has been deeply reviewed and completed to be closer to the specifications and to implement the best security practices. As a result, the Nuxeo Mobile application (/nxdoc/nuxeo-mobile/) will now use the OAuth flow to access content stored in the Nuxeo repository.

➡ More on JIRA epic NXP-21721 (https://jira.nuxeo.com/browse/NXP-21721).

PKCE For Native Applications

Starting from Nuxeo 9.3, security has been improved on the OAuth flow implemented by Nuxeo Server, so that applications trying to connect to Nuxeo via a redirect to a custom URI scheme can be validated as non-malicious. Typically since several applications could be listening to such scheme (ex: `nuxeo://`).

➡ More on JIRA ticket NXP-22193 (https://jira.nuxeo.com/browse/NXP-22193).

RFC Recommendations about OAuth2 Response Errors and Log These Error

Error responses happening during the OAuth flow have been improved so as to stick to the RFC:

- Authorization response: https://tools.ietf.org/html/rfc6749#section-4.1.2.1 (https://tools.ietf.org /html/rfc6749#section-4.1.2.1)
- Access token response: https://tools.ietf.org/html/rfc6749#section-5.2 (https://tools.ietf.org /html/rfc6749#section-5.2)

Errors are also logged server-side.

➡ More on JIRA ticket NXP-22717 (https://jira.nuxeo.com/browse/NXP-22717).

REST API Endpoints for OAuth Tokens and OAuth Providers Management

Endpoints to CRUD OAuth tokens and providers (in respect to permission) have been added:

- `GET oauth2/provider` : Returns the list of registered OAuth2 service providers
- `POST oauth2/provider` : Creates a new OAuth2 service provider
- `PUT oauth2/provider/<providerName>` : Updates an OAuth2 service provider
- `DELETE oauth2/provider/<providerName>` : Deletes an OAuth2 service provider
- `GET oauth2/token` : Gets the list of all stored tokens for all users
- `GET oauth2/token/<providerName>/<nxLogin>` : Retrieves a stored token

- `PUT oauth2/token/<providerName>/<nxLogin>` : Updates a stored token
- `DELETE oauth2/token//<providerName>/<nxLogin>` : Deletes a stored token

➡ More on JIRA ticket NXP-20884 (https://jira.nuxeo.com/browse/NXP-20884).

Transient Store

Transient Store Based on Key/Value Service and Blob Provider

The transient store has been re-implemented on top of the KeyValueService and a BlobProvider so as to remove the dependency on Redis (when no service makes use of it anymore).

➡ More on JIRA ticket NXP-21871 (https://jira.nuxeo.com/browse/NXP-21871).

REST API

Asynchronous Download

A new asynchronous download mechanism has been introduced. It allows getting a status of the asynchronous preparation of the blob to download, before downloading it for real. For instance when calling the operation `Blob.BulkDownload` answer will first be:

```
"key":"4db11225407dad8432e75ba2d2778a49","completed":false,"progress":0
```

Then calling `blobstatus` enpoint with the provided key `https://nightly.nuxeo.com/nuxeo/nxblobstatus/4db11225407dad8432e75ba2d2778a49` gives an updated result:

```
{"key":"4db11225407dad8432e75ba2d2778a49","completed":true,"progress":100}
```

When completed, one can call the download servlet with the same key:

```
{
https://nightly.nuxeo.com/nuxeo/nxbigblob/4db11225407dad8432e75ba2d2778a49
}
```

➡ More on JIRA ticket NXP-23062 (https://jira.nuxeo.com/browse/NXP-23062).

Add a Comment While Updating a Document Through the REST API

An update comment can now be passed through the `Update-Comment` header when updating a document through the REST API. Comment will be added to the Audit log.

➡ More on JIRA ticket NXP-23319 (https://jira.nuxeo.com/browse/NXP-23319).

Jersey Client Handler

REST API Client handler now uses Apache's Jersey implementation while it was using the JDK one previously. It provides stricter behaviors leading to better resilience.

➡ More on JIRA ticket NXP-22798 (https://jira.nuxeo.com/browse/NXP-22798).

documentURL Enricher is Adapted to Using Web UI

`documentUrl` enricher now takes into account the application that has been set as the default one (between JSF an Web UI) so as to set the correct base on the returned.

➡ More on JIRA ticket NXP-22875 (https://jira.nuxeo.com/browse/NXP-22875).

User Management

nuxeo.user.emergency.enable Disabled By Default

Virtual emergency user that was available by default when using LDAP for authentication is disabled by default for security reason.

➡ More on JIRA ticket NXP-22888 (https://jira.nuxeo.com/browse/NXP-22888).

Improve LDAP directory Performance When Using Cache

LDAP connector has been optimized.

➡ More on JIRA ticket NXP-19848 (https://jira.nuxeo.com/browse/NXP-19848).

Migration Service

New Migration Service

A new migration service defines a series of states that a migration can take and allows service aware of this migration to read and write accordingly in the former or new way, depending on the status of the migration. The goal of this service is to standardize how future migrations will be processed within the Nuxeo Platform. The first use case of this new service is the migration of the Tag Service implementation.
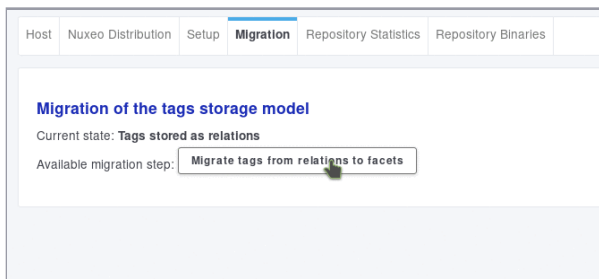
→ More on JIRA ticket NXP-19412 (https://jira.nuxeo.com/browse/NXP-19412).

Packaging

Tomcat 8.5.23

Tomcat Server has been upgraded to version 8.5.23

→ More on JIRA ticket NXP-20370 (https://jira.nuxeo.com/browse/NXP-20370).

Improved Tomcat Configurability

More configuration is allowed on the embedded Tomcat HTTP pool: number of HTTP threads and queue size. This allows to avoid DOS. We also give in server.xml an example of configuration with two HTTP pools, one for Drive one for other web requests.

→ More on JIRA ticket NXP-23318 (https://jira.nuxeo.com/browse/NXP-23318).

Static WAR Generation

Static WAR generation has been fixed and added back to continuous integration. Nuxeo static WAR can be deployed as a standard web application on a Tomcat server.

→ More on JIRA ticket NXP-21659 (https://jira.nuxeo.com/browse/NXP-21659).

User workspace

More Robust User Workspace Name Definition

We now generate the user workspace name based on the user id with no possible collision. We use an escaping mechanism that escapes the few unwanted characters in paths and URLs (slash, backslash, `?`, `&`, `@`, `&`) using a `~` and an hex code. This applies only to new user workspaces; If a user workspace was created previously, it will still be recognized and used.

→ More on JIRA ticket NXP-21807 (https://jira.nuxeo.com/browse/NXP-21807).

Addons

Packaging

Addons' Registries

Some of the Nuxeo packages now include the list of contributions that they deploy in a JSON registry following the same format than the one expected for Nuxeo Studio registries.

➜ More on JIRA ticket NXP-23310 (https://jira.nuxeo.com/browse/NXP-23310).

Lambda Integration

A new addon has been implemented that allows to leverage Amazon Lambdas for efficient asynchronous computing. The initial use case is to offload the Picture Views generation (generating several renditions with different sizes of the same source image), but the module has been designed to be used for other kinds of processing.

➜ More on the readme (https://github.com/nuxeo/nuxeo-lambda/blob/master/Readme.md) and on the JIRA ticket NXP-21922 (https://jira.nuxeo.com/browse/NXP-21922).

Imaging

Disabling Default Picture Conversions

Default picture conversions can now be disabled by using the `enabled` attribute on the PictureConversions extension point (http://explorer.nuxeo.com/nuxeo/site/distribution/latest/viewExtensionPoint/org.nuxeo.ecm.platform.picture.ImagingComponent--pictureConversions) contributions.

➜ More on JIRA ticket NXP-21311 (https://jira.nuxeo.com/browse/NXP-21311).

New Layout For Picture Document Type

A new layout has been implemented for pictures so as to display all picture related metadata (IPTC, technical information, etc.).
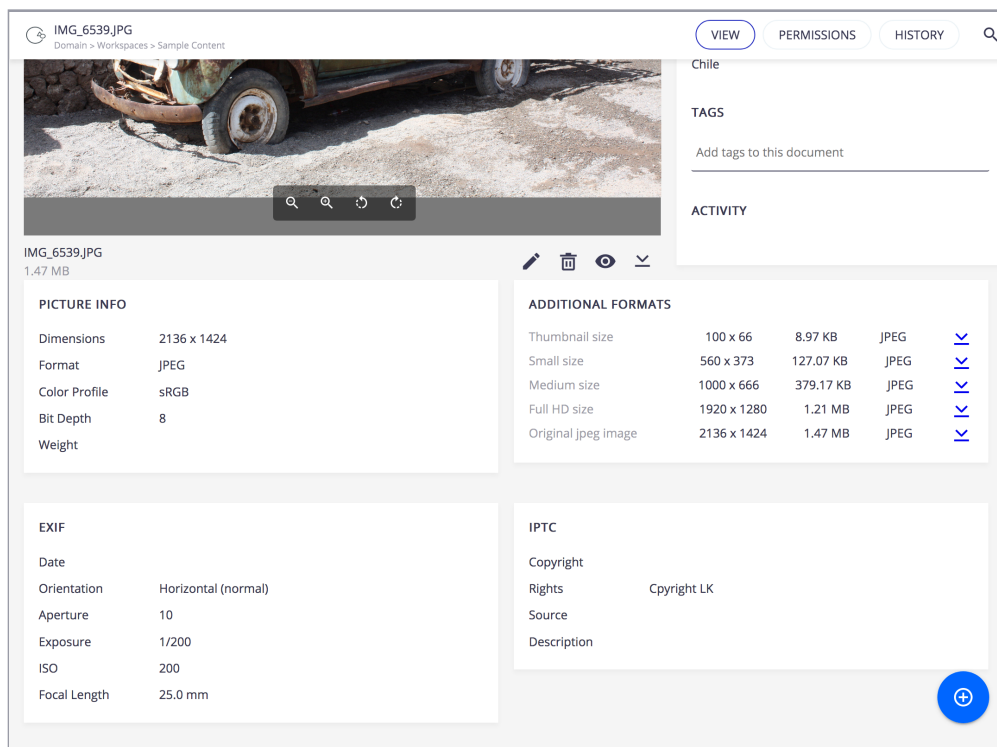
➡ More on JIRA ticket NXP-23064 (https://jira.nuxeo.com/browse/NXP-23064).

Binary Metadata

Binary metadata (/nxdoc/binary-metadata/) allows to extract and write back metadata on binaries (ex: EXIF metadata). The module no more tries to update the binary when the content is managed externally, for instance when using Nuxeo Live Connect (/nxdoc/nuxeo-live-connect/).

➡ More on JIRA ticket NXP-20282 (https://jira.nuxeo.com/browse/NXP-20282).

Nuxeo Drive

Server-Side Configuration Of Drive Clients

Some properties of the Drive clients can now be configured (and updated) from the server for all Drive clients:

```
{
"delay": 30,
"ignored_prefixes": [".", "icon\r", "thumbs.db", "desktop.ini", "~$"],
"ignored_suffixes": [".bak", ".crdownload", ".lock", ".nxpart", ".part",
".partial", ".swp", ".tmp", "~", ".dwl", ".dwl2"],
"ignored_files": ["^atmp\\d+$"],
"log_level_file": "DEBUG",
"timeout": 30,
"handshake_timeout": 60,
"beta_channel": false,
"update_check_delay": 3600,
"ui": "web"
}
```

Do not hesitate to give some feedback if you see some additional properties that would be useful to add here.

Do Not Copy Registrations When Copying a Folder

Root registrations are not copied when a document and its children are copied. In 9.1 and later, by default, we reset the synchronization root registrations on a copied document and its children. In LTS 2016 and earlier the previous behavior is kept.

➡ More on JIRA ticket NXP-21676 (https://jira.nuxeo.com/browse/NXP-21676).

Nuxeo Vision

Amazon Rekognition

Nuxeo Vision (/nxdoc/nuxeo-vision/), the computer vision gateway, can now use Amazon Rekognition in addition to Google Vision. The automation operation `VisionOp` now includes a new parameter `provider` that allows to specifically call a given provider.

➡ More on JIRA ticket NXP-21851 (https://jira.nuxeo.com/browse/NXP-21851).

Live Connect

Dropbox API V2 Implementation

Nuxeo Live Connect (https://connect.nuxeo.com/nuxeo/site/marketplace/package/nuxeo-liveconnect) now leverages the Dropbox API v2.

➡ More on JIRA ticket NXP-22384 (https://jira.nuxeo.com/browse/NXP-NXP-22384).

Nuxeo CSV

Nuxeo CSV Web UI Port

Nuxeo CSV (/nxdoc/nuxeo-csv/) can now be used from Web UI (/userdoc/web-ui/). The feature is available from the import pop-up, in a new "CSV" tab.
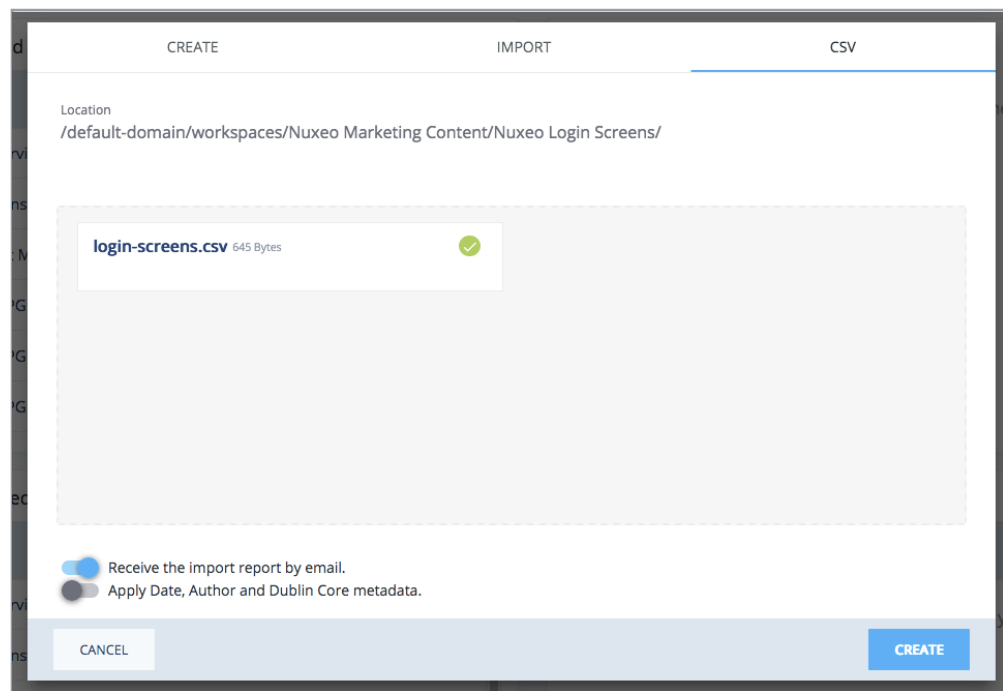
➜ More on JIRA ticket NXP-21484 (https://jira.nuxeo.com/browse/NXP-21484).

CSV Importer Report

The Web UI integration of CSV importer now displays the import report to the user.

➜ More on JIRA ticket NXP-22460 (https://jira.nuxeo.com/browse/NXP-22460).

Handle Blobs in Lists and Complex Types

By using a JSON format like

```
[
    {
        "file": {
            "mime-type": "text/plain",
            "content": "first_file.txt"
        }
    },
    {
        "file": {
            "mime-type": "application/pdf",
            "content": "my_pdf.pdf"
        }
    },
    {
        "file": {
            "name": "custom_name.pdf",
            "content": "another_pdf.pdf"
        }
    }
]
```

we can now import files in the attachment part of the File document (or any of your customized document type), while only the main `file:content` property could be filled using Nuxeo CSV importer before.

➜ More on JIRA ticket NXP-22746 (https://jira.nuxeo.com/browse/NXP-22746) and on the documentation (/nxdoc/nuxeo-csv/#files-attachment)

Nuxeo Template Rendering Web UI Port

The template Rendering plugin (/nxdoc/template-rendering-addon/) has been ported to Web UI. It has also been simplified. It is now designed to be used mostly in some situations where the Web UI has been tailored to your use cases. A user action looks up for matching templates and lets user choose one if there are several. If the template has some parameters, that action will asks the user those parameters, otherwise it will generate the rendition and fire a download. It is possible to easily override the template look up logic as well as the rendition chain that is used.
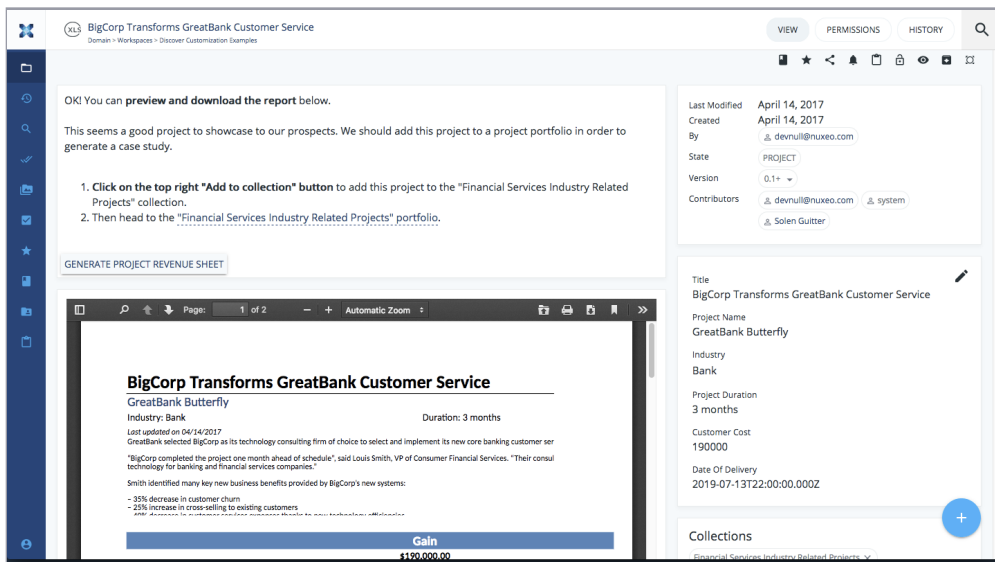
➜ More on JIRA ticket NXP-21486 (https://jira.nuxeo.com/browse/NXP-21486).

Nuxeo Platform Importer

English Dictionary for Random Import

Random English content can now be generated using the random importer.

➜ More on JIRA ticket NXP-21260 (https://jira.nuxeo.com/browse/NXP-21260).

Nuxeo Importer Stream   NEW

A new addon has been added, called "nuxeo-importer-stream", that exposes a consumer / producer pattern making use of the nuxeo-stream infrastructure for running massively scalable imports. See the readme of the addon (https://github.com/nuxeo/nuxeo-platform-importer/tree/master/nuxeo-importer-stream).

➜ More on JIRA ticket NXP-23560 (https://jira.nuxeo.com/browse/NXP-23560).

Nuxeo EasyShare Port

EasyShare addon (/nxdoc/easyshare/) has been ported to Web UI.
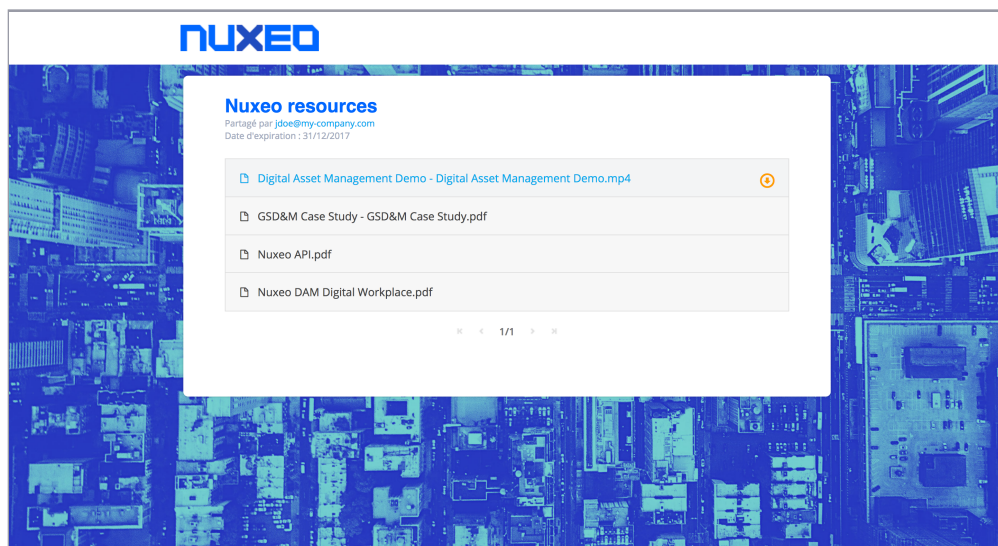
➡ More on JIRA ticket NXP-22923 (https://jira.nuxeo.com/browse/NXP-22923).

Quota

The quota limit used to be hardcoded to 999GB. It is now configurable by contributing an extension like:

```
<require>org.nuxeo.ecm.quota.maxsize.config</require>
  <extension target="org.nuxeo.runtime.ConfigurationService"
point="configuration">
    <property name="nuxeo.quota.maxsize">2 TB</property>
  </extension>
```

The size is expressed in bytes but can use a suffix like KB, MB, GB or TB.

➡ More on JIRA ticket NXP-22702 (https://jira.nuxeo.com/browse/NXP-22702).

Nuxeo Web UI

Nuxeo Web UI Becomes the Default UI

With Nuxeo Platform 9.3 Nuxeo Web UI becomes the default UI of the Platform.

➡ More on JIRA ticket NXP-23087 (https://jira.nuxeo.com/browse/NXP-23087).

Improved Performance

Loading time of Nuxeo Web UI on Firefox, Internet Explorer and Edge has been improved.

➡ More on JIRA ticket NXP-20662 (https://jira.nuxeo.com/browse/NXP-20662).

New List Item View with Search Results Highlight

A new search result view has been added on the main search. It displays items on the full width of the panel, in the same style as a web search engine such as Google, with searched terms highlighted in the context of the original text wrapping them.
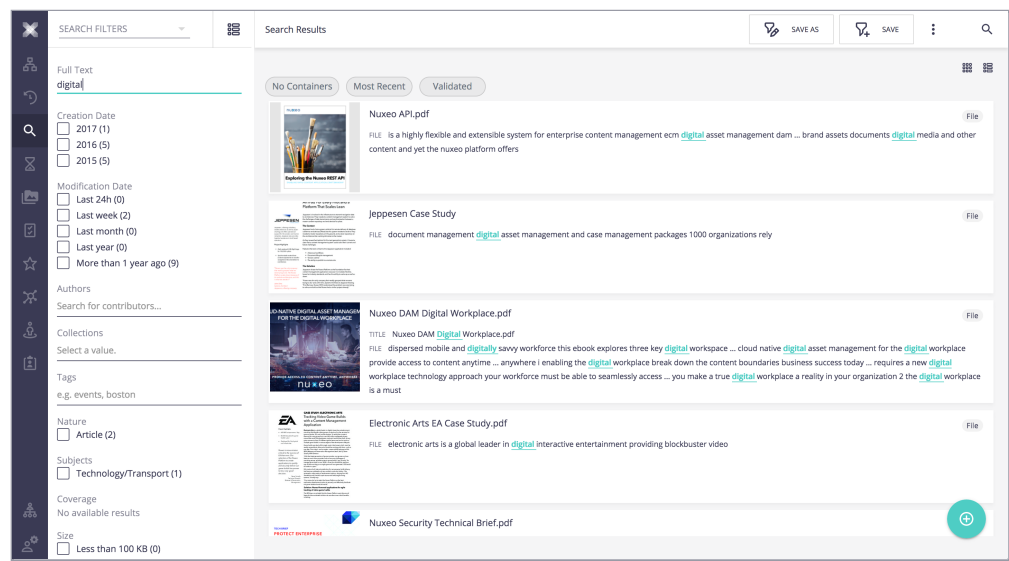
➜ More on JIRA ticket NXP-22567 (https://jira.nuxeo.com/browse/NXP-22567).

Results Highlight in Suggestion Search

When doing a quick search in the suggestion box, Web UI now displays the matched terms and metadata so as to understand why the result item is returned.
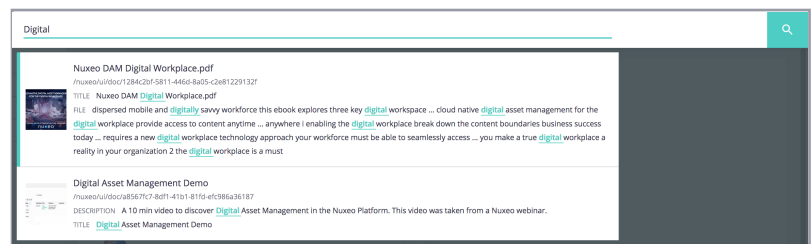
➜ More on JIRA ticket NXP-22566 (https://jira.nuxeo.com/browse/NXP-22566).

Justified Thumbnails Grid

A new view for media search results has been added, with justified display of thumbnails that keep their original ratio and metadata displayed on hover. The goal is to improve user experience for users that spend a lot of time searching for visual media (images, video) in the repository.
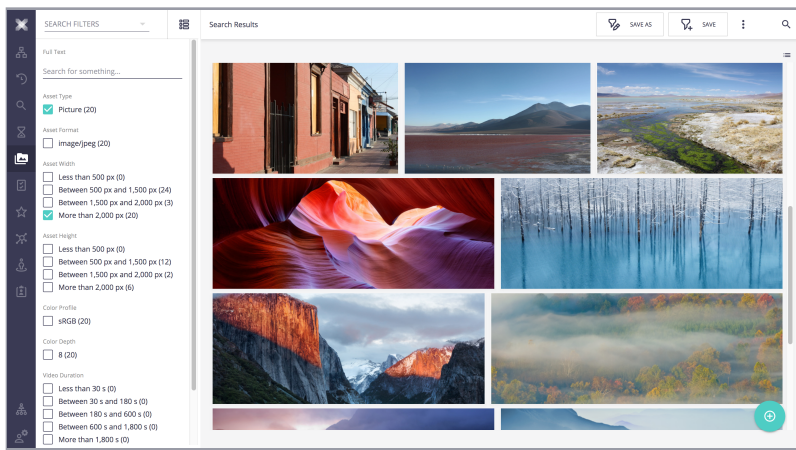
➜ More on JIRA ticket NXP-22474 (https://jira.nuxeo.com/browse/NXP-22474).

Lazy Scrolling

It is now possible to easily and immediately scroll a whole result set of documents / images / videos. When the result set is sorted, the default UI provides ability to scroll up to a certain aggregate value, facilitating the ability to find the exact image that was required.

➜ More on JIRA ticket NXP-22668 (https://jira.nuxeo.com/browse/NXP-22668).

Bulk Download Action on Document Selection

A new action is available in the bulk selection actions list, that allows to bulk download selected content in a single ZIP downloaded asynchronously. The main file is added to the ZIP for each selected document. When it comes to folders, content is recursively resolved up to a configurable level.
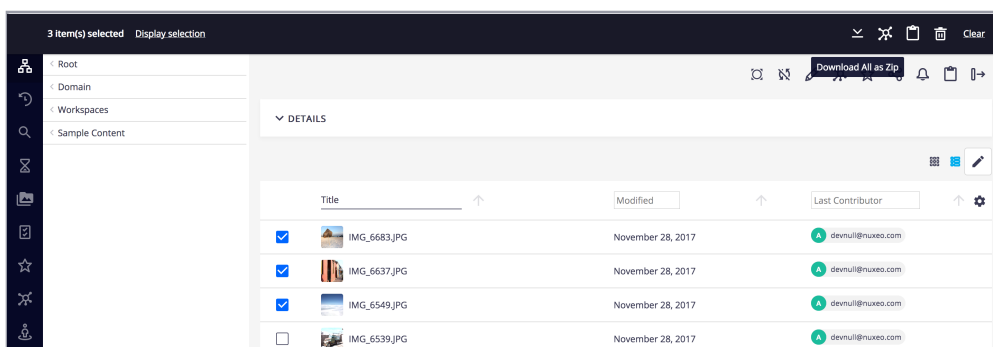
➜ More on JIRA ticket NXP-23095 (https://jira.nuxeo.com/browse/NXP-23095).

Improved Task View

Task view has been improved:

- New task layout: More readable task details, preview on the right, flatter design
- Task queue: Optimization of the information available to the user
- Tasks Listing Dashboard: File name and path info have been added
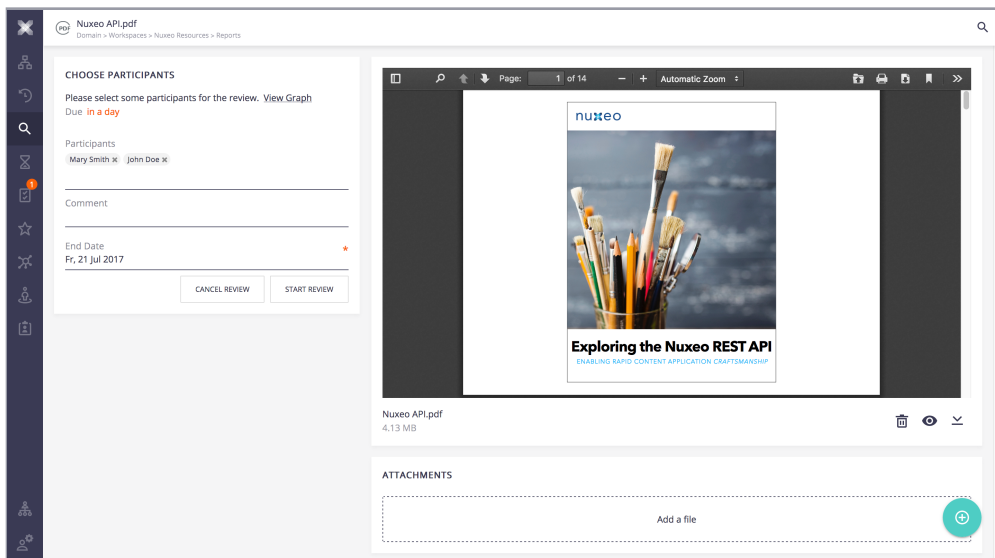
➜ More on JIRA ticket NXP-22591 (https://jira.nuxeo.com/browse/NXP-22591).

New Layout For Folderish Documents

The folder view (and view of other folderish documents) has been improved so that more space is dedicated to the content listing of the folder while displaying metadata of the folder above the listing.
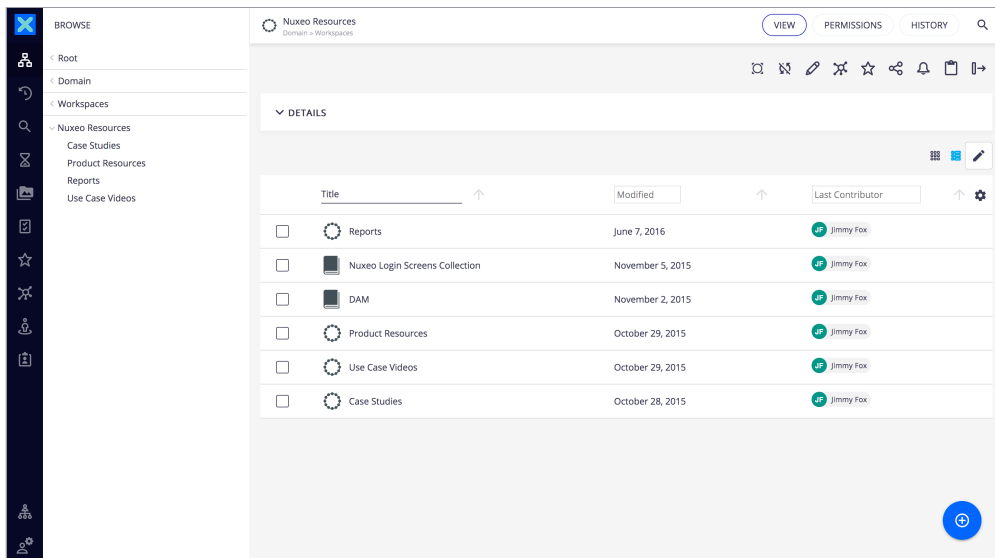
➜ More on JIRA ticket NXP-22997 (https://jira.nuxeo.com/browse/NXP-22997).

Better Document Edit User Experience

More space has been dedicated to metadata edit experience, by using a pop-in that displays metadata edit page full screen.

→ More on JIRA ticket NXP-22989 (https://jira.nuxeo.com/browse/NXP-22989).

Storyboard on Video Layout

The storyboard has been added to Web UI view of videos.

→ More on JIRA ticket ELEMENTS-358 (https://jira.nuxeo.com/browse/ELEMENTS-358)

NXQL Search For Administator Users

The Nuxeo Web UI Admin menu now shows a NXQL search menu, enabling administrators to query documents in NXQL for more flexibility.

➡ More on JIRA ticket NXP-22766 (https://jira.nuxeo.com/browse/NXP-22766).

Polymer 2.0

Nuxeo Elements now depends on Polymer 2.0.

➡ More on JIRA ticket NXP-23100 (https://jira.nuxeo.com/browse/NXP-23100).

nuxeo-result-view

A new element can be used to display the result of a search in any page, based on the query of a page provider. The element is called `nuxeo-results-view`.

➡ More on JIRA ticket NXP-22679 (https://jira.nuxeo.com/browse/NXP-22679).

Error Message On Elements

The following elements now enable to display a custom message when the value is incorrect:

* nuxeo-user-suggestion
* nuxeo-directory-suggestion
* nuxeo-document-suggestion
* nuxeo-tag-suggestion
* nuxeo-date-picker
* nuxeo-input
* nuxeo-textarea

➡ More on JIRA ticket ELEMENTS-353 (https://jira.nuxeo.com/browse/ELEMENTS-353).

Default Workflows Independent from Nuxeo JSF UI

Default Workflows (serial and parallel reviews) have been added to Nuxeo Web UI and do not require Nuxeo SJF UI anymore.

➡ More on JIRA ticket NXP-21518 (https://jira.nuxeo.com/browse/NXP-21518).

More Style Variables For Further Customization

More CSS variables area available allowing to move some of the major elements (like search bar) and to better control the space available for the logo.

➡ More on JIRA ticket NXP-23198 (https://jira.nuxeo.com/browse/NXP-23198).

Add an Alternative Text to Non-text Content

Some improvements in regard to accessibility of the Web UI application have been implemented, like adding an alternative text to non-text content as well as adding relevant information to page title dynamically.

� More on JIRA tickets NXP-22478 (https://jira.nuxeo.com/browse/NXP-22478) and NXP-22477 (https://jira.nuxeo.com/browse/NXP-22477).

Ability to Close the Banner

A close button has been added to the Mobile Banner in Web UI.

➤ More on JIRA ticket NXP-21794 (https://jira.nuxeo.com/browse/NXP-21794).

Ability to Customize Tool-Tip Position in Document Actions

The tool-tip position attribute allows configuring the position of the tool-tip on most of our actions.

➤ More on JIRA ticket NXP-23207 (https://jira.nuxeo.com/browse/NXP-23207).

Hungarian Is Now Translated Using Crowdin

➤ More on JIRA ticket NXP-22758 (https://jira.nuxeo.com/browse/NXP-22758).

Nuxeo JSF UI

Custom Mobile Banner
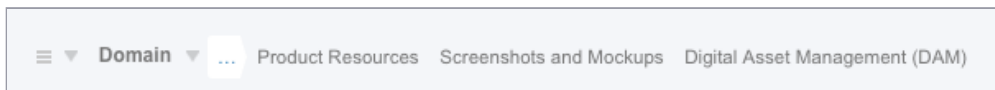
A Custom Banner has been added so as to let the user open the Nuxeo mobile application (/nxdoc /nuxeo-mobile/) when browsing the JSF UI from an Android or iOS mobile device.
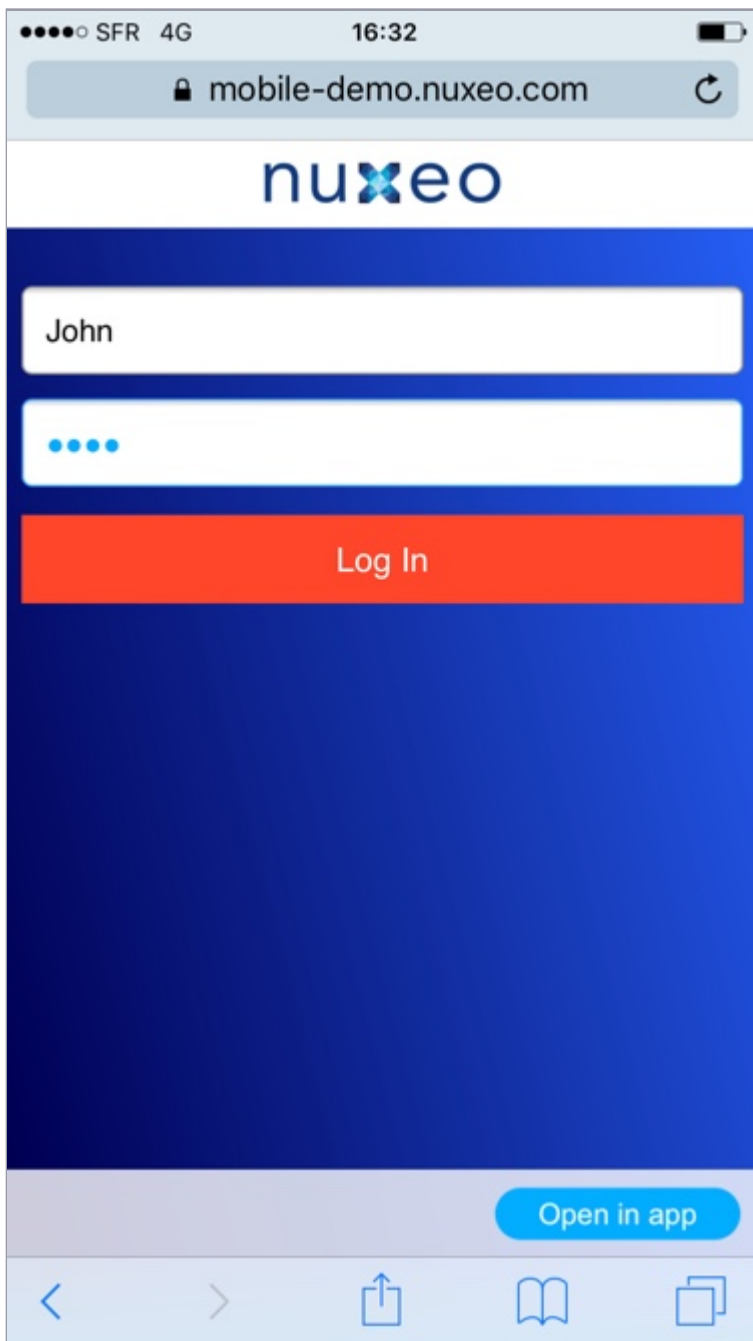
→ More on JIRA ticket NXP-21679 (https://jira.nuxeo.com/browse/NXP-21679).

Breadcrumb: Browse Parent Folder

Dots in the breadcrumb allow to move to parent folder.

→ More on JIRA ticket NXP-21068 (https://jira.nuxeo.com/browse/NXP-21068).

Integer Constraint Correctly Formatted in JSF UI

When a property is of type Integer and the constraint check is enabled, message on the widget is:
"'abc' is not a number. Example: 99".

New Access Simple Access URL for JSF Application

When the nuxeo-jsf-ui package is installed, the JSF UI can now be accessed via `http://SERVER_URL/nuxeo/jsf` . This enables an easy access to Nuxeo JSF UI even though Nuxeo Web UI is now the default UI of the Platform.

➡ More on JIRA ticket NXP-23376 (https://jira.nuxeo.com/browse/NXP-23376).

Future Deprecation of Nuxeo JSF UI

Nuxeo JSF UI (https://connect.nuxeo.com/nuxeo/site/marketplace/package/nuxeo-jsf-ui) will be deprecated and released one last time for LTS 2018, in one year. The end of support is scheduled for end of 2020. Nevertheless, we recommend that you start your new projects based on Nuxeo Platform LTS 2017 using our latest UI, Nuxeo Web UI (https://connect.nuxeo.com/nuxeo/site/marketplace/package/nuxeo-web-ui). It brings you:

- More insurance of browser compatibility in the future
- A more responsive application
- More scalability
- A better UX
- A guarantee of new features coming over the years, while JSF application developments are frozen

Metrics

Filtering Metrics

A mechanism to authorize some metrics and to filter some from being sent to graphite has been added.

```
<extension target="org.nuxeo.runtime.metrics.MetricsService"
point="configuration">
<graphiteReporter enabled="true" periodInSecond="20"
host="graphite.server.name" port="2003"
prefix="servers.${hostname}.nuxeo.">
<allowedMetrics>
<metric>nuxeo.cache.user-entry-cache</metric>
<metric>nuxeo.cache.group-entry-cache</metric>
</allowedMetrics>
<deniedMetrics>
<metric>nuxeo.cache</metric>
<metric>nuxeo.directories</metric>
</deniedMetrics>
</graphiteReporter>
</extension>
```

➜ More on JIRA ticket NXP-22994 (https://jira.nuxeo.com/browse/NXP-22994).

JSF to Web UI Addon   N E W

This new addon available on the Marketplace (https://connect.nuxeo.com/nuxeo/site/marketplace/package/nuxeo-jsf-to-web-ui) allows redirecting JSF URLs to Web UI ones, especially the permanent links. This is useful if you have used the JSF application and now use Web UI and that users have bookmarked some JSF links.

➜ More on JIRA ticket NXP-23440 (https://jira.nuxeo.com/browse/NXP-23440).

New Customer Plugin Sample

A new sample project is available that follows customer's plugin good practice and close to your needs. This sample:

- Is generated/Reproducible from Nuxeo CLI
- Has a Nuxeo Platform independent versioning
- Holds a Nuxeo Package in the same module as the source code

➜ More on JIRA ticket NXP-22891 (https://jira.nuxeo.com/browse/NXP-22891).

WebEngine

Meaningful HTTP Status Code

`WebException` (and most subclasses) has been deprecated, `NuxeoException` (and subclasses) are now the exceptions to be thrown, and those can be marshalled correctly by the new registered `JsonNuxeoExceptionWriter`.

A new `statusCode` field has been added to `NuxeoException` to specify which HTTP code should be returned in case the exception is thrown, default to 500.

`ModuleResource#handleError(WebApplicationException e)` has been changed to `ModuleResource#handleError(Throwable t)` to handle error at the WebEngine module level.

Compatibility has been kept to still write `WebException` as JSON if they are thrown.

➜ More on JIRA ticket NXP-21776 (https://jira.nuxeo.com/browse/NXP-21776).

Packaging

Official Docker Image is Compatible with OpenShift

The script to build the official Docker image has been improved so that the image can be launched from Open Shift, which imposes some specific security constraints.

→ More on JIRA ticket NXP-21880 (https://jira.nuxeo.com/browse/NXP-21880).

CentOS Specific Image Build

The script to build the official Docker image now also allows to build specifically an image on top of CentOS.

→ More on JIRA ticket NXP-21995 (https://jira.nuxeo.com/browse/NXP-21995).

`nuxeoctl register` new command

Two new `nuxeoctl` commands are available to make it easier to manage your license from the command line:

- `nuxeoctl register --clid 9e7e0b7a-3e75-4e79….` : allows to install a clid offline
- `nuxeoctl register —renew` : allows to renew your license

→ More on JIRA ticket NXP-21564 (https://jira.nuxeo.com/browse/NXP-21564).

Farewell

Nuxeo Forum

Deprecated for LTS 2016, Nuxeo Forum has been removed.

→ More on JIRA ticket NXP-21448 (https://jira.nuxeo.com/browse/NXP-21448).

Box API

Deprecated for LTS 2016, Box API has been removed.

→ More on JIRA ticket NXP_22041 (https://jira.nuxeo.com/browse/NXP-22041).