

Nuxeo Server 10.2 Fast Track Release Notes

VERSION: FT

Updated: July 26, 2018

This page relates to the release notes of Nuxeo Server and related addons for the 10.10 cycle, a.k.a LTS 2018 cycle. It will list the improvements and features that are successively shipped with the 10.1, 10.2, 10.3 and LTS 2018 releases. Evolutions are grouped by components.

You can also find detailed JIRA release notes:

- [10.1 JIRA release notes](#)
- [10.2 JIRA release notes](#)

We also provide [instructions for upgrading](#) to the latest release.

Nuxeo Server

Core Repository

Trash, Untrash and EmptyTrash Operations

Two new

operations `TrashDocument` and `UntrashDocument` have been added.

More on JIRA ticket [NXP-24282](#) and [NXP-24281](#).

New Trash Service Enabled

The new trash service has been enabled by default on the repository. It uses a system property `ecm:isTrashed` for labelling a document as being trashed. It also fires dedicated events `documentTrashed` and `documentUntrashed` (hold by TrashService interface). A migrator has been implemented for migrating content from the old trash system (relying on lifecycle state) to the new one (relying on the system property `ecm:isTrashed`). See upgrade notes for more information on migration.

More on JIRA tickets [NXP-24850](#) and [NXP-24035](#)

ecm:isTrashed in the JSON

`ecm:isTrashed` is now in the JSON representation of a Nuxeo Document

More on JIRA ticket [NXP-24741](#)

New firstAccessibleAncestor REST API Enricher

It is now possible to get the closest document's ancestor of a document using the `firstAccessibleAncestor` JSON Enricher.

More on JIRA ticket [NXP-24282](#)

No Mention of the Repository on a Document Reference

When referencing a document in a property, we don't need anymore to store the repository id. If the repository id is not there, the same as the referencing document is chosen. The following two restrictions with and can be used for this:

```
<xs:restriction base="xs:string"  
ref:resolver="documentResolver" ref:store="idOnly" />
```

```
<xs:restriction base="xs:string"  
ref:resolver="documentResolver"  
ref:store="pathOnly" />
```

Their semantics is to store only the id or only the path, without any prefixed repository. When fetching the constraint, the document of the given id or path is resolved in the same repository as the current document.

More on JIRA ticket [NXP-22450](#).

GetProxies Method on CoreSession

`getProxies(DocumentRef)` has been added to the CoreSession to efficiently get proxies, without having to write and run a query.

More on JIRA ticket [NXP-24922](#).

KeyValueStoreUIDSequencer

A new `KeyValueStoreUIDSequencer` is available, to store sequences in a key/value store. The store is the same for all sequencers, but they are using different keys, prefixed by the sequencer name.

It can be used by doing for example:

```
<extension
target="org.nuxeo.ecm.core.uidgen.UIDGeneratorService
" point="sequencers">
  <sequencer name="uidgen"
class="org.nuxeo.ecm.core.uidgen.KeyValueStoreUIDSequ
encer" default="true" />
</extension>
```

```
<extension
target="org.nuxeo.runtime.ConfigurationService"
point="configuration">
  <!-- this is an example, "sequence" is already the
default -->
  <property
name="nuxeo.uidseq.keyvaluestore.name">sequence</
property>
</extension>
```

Assuming the following configuration to define a suitable key/value store, for example:

```
<extension
target="org.nuxeo.runtime.kv.KeyValueService"
point="configuration">
  <store name="sequence"
class="org.nuxeo.ecm.core.mongodb.kv.MongoDBKeyValues
tore">
  <property name="collection">sequence</property>
</store>
```

</extension>

More on JIRA ticket [NXP-23744](#).

Core Storage

KMS Keys Support on S3 Binary Store

The support for [KMS keys](#) for S3 Server-Side Encryption is added.

More on JIRA ticket [NXP-22949](#).

Directory

Tenant-isolated directories with MongoDB

Multi-tenant add-on now supports tenant-isolated directories with MongoDB

More on JIRA ticket [NXP-22682](#)

Same Directory Entry ID on Different Tenants

Unicity check on directory entry has been moved post tenant-specific computation, so that same end user id can be used in two different tenants.

More on JIRA ticket [NXP-25264](#)

LDAP Connection timeout reduced to 1 min

The pooling connection timeout of the LDAP connections has been reduced from 30 min to 1 min to avoid some performance issues when making a lot of connections with multiple users.

Multi-tenant addon now supports tenant-isolated directories with MongoDB More on JIRA ticket [NXP-25085](#)

Filters parameter on Directory.SuggestEntries

It is now possible to filter directories values on a given column value, so as to implement chain select behaviors for instance. You can use the "filters" parameter, with a serialized property,

like: `{"parent": "europe"}`

[NXP-25299](#)

Workflow

More Properties on the Task Object

When using the REST API, the JSON structure of a Task object now also includes:

- the workflow initiator
- the workflow title
- the workflow life cycle state
- the graph route URL

More on JIRA ticket [NXP-24476](#).

Nuxeo Streams

Avro serialization for Nuxeo Log/Stream

Avro (<http://avro.apache.org>) has been integrated to the platform: Avro schema store, Nuxeo Document <> Avro format converter so as to be able to use this format for Nuxeo Stream messages. It makes communication with outer world easier (no Nuxeo dependency in the message), makes messages more compact and give backward / forward compatibility in messages format. As a consequence, Nuxeo Stream can now encode record with different codec:

- `legacy`: the original format based on java Externalizable
- `avro`: avro message with a schema fingerprint header (Nuxeo has an avro SchemaStore service to retrieve schemas).
- `avroBinary`: avro message without schema header so more compact
- `avroJson`: avro in JSON for debug purpose only

You can choose the encoding for the different service using `nuxeo.conf` options:

```
nuxeo.stream.work.log.codec=legacy
```

`nuxeo.stream.audit.log.codec=legacy`

`nuxeo.stream.pubsub.log.codec=avroBinary`

Note that you should not change the codec of an existing stream (Kafka Topic or Chronicle file), this should be done only on new stream.

More on JIRA ticket [NXP-22597](#) and [NXP-24324](#)

Kafka 1.1

The Nuxeo Platform now relies on Kafka 1.1.

More on JIRA ticket [NXP-25087](#).

Stream Computations Use Watermark

Computations that re-append records in other streams now add a watermark (time and flag information) in the record. That allows always benefiting from advanced features of the streams such as latency computation.

More on JIRA ticket [NXP-24641](#)

Using The Key as the Kafka Record Key

When using the Kafka implementation of Nuxeo Stream, the key in the log is now the Kafka key itself, not the hash of it as it was previously (only option for the other backend, Chronicle queue).

More on JIRA ticket [NXP-24640](#).

WorkManager

Error Event After Successive Failures on a Work

An event `workFailed` is now fired when a work fails several times.

More on JIRA ticket [NXP-24126](#).

State Information for the StreamWorkManager

By using `nuxeo.stream.work.storestate.enabled=true`, it is now possible to get information of a given work when using the StreamWorkManager.

More on JIRA ticket [NXP-24397](#).

Ability to Cancel a Long-running Work with the StreamWorkManager

StreamWorkManager is now able to cancel a long-running work, provided the work calls regularly `isSuspending`.

More on JIRA ticket [NXP-24400](#).

Scheduler

Customisable Job

It is now possible to specify what Job should be created when using the scheduler thanks to a new attribute `jobFactoryClass`:

```
<schedule id="testSchedulerSingleExecution"
```

```
jobFactoryClass="org.nuxeo.ecm.core.scheduler.SingleE
xecutionEventJobFactory">
  <event>testSchedulerSingleExecution</event>
  <cronExpression>* * * * * ?</cronExpression>
</schedule>
```

More on JIRA ticket [NXP-24540](#).

Timezone Sensitive Cron Service

The scheduler service (Cron) now allows using a timezone on expressions to avoid relying on the time of the server for firing events.

More on JIRA ticket [NXP-24955](#).

PubSub

PubSub Service on Nuxeo Stream

An implementation of the PubSub service has been provided using Nuxeo Stream. This allows to not rely on Redis for this service that is notably used for cache syncing on the repository in a cluster, as well as for acquiring locks on documents. To use it you can apply the following configuration in [nuxeo.conf](#):

```
nuxeo.pubsub.provider=stream
```

More on JIRA ticket [NXP-23799](#).

Audit

STARTSWITH Operator Available for All Audit Backend (SQL, MongoDB and Elasticsearch)

Following introduction

of `AuditBackend#queryLogs(AuditQueryBuilder)`, we now have an easy way to query audit. We introduced in 10.1 the STARTSWITH operator, we could use it as below:

```
auditBackend.queryLogs(new
AuditQueryBuilder().predicates( //
    Predicates.eq(LOG_EVENT_ID,
" SOMETHING" ),
    Predicates.startsWith(LOG_DOC_PATH,
"/myFolder" ))) ;
```

More on JIRA ticket [NXP-24396](#).

Query

NXQL `ecm:isTrashed` Support

Following evolutions on the trash service, the NXQL property `ecm:isTrashed` has been added to be able to filter queries on trashed or not trashed documents.

Some Built-In Page Providers Moved to Elasticsearch

`REST_API_SEARCH_ADAPTER` and `all_collections` page providers have been added to the default list of page providers

provided by Elasticsearch. If you have defined your own `elasticsearch.override.pageproviders` then it is recommended to add those two to your list.

More on JIRA ticket [NXP-24346](#).

Elasticsearch

Elasticsearch 6.3

Elasticsearch 6.3+ is now required. See upgrade instructions.

More on JIRA ticket [NXP-24102](#).

Support of X-Pack

The use of Elasticsearch X-Pack is now allowed, [see documentation](#).

More on JIRA ticket [NXP-23048](#).

More Like This Hint

A new hint is available that allows to leverage the More Like This query of Elasticsearch (<https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-mlt-query.html>)

Ex:

```
SELECT * FROM Document WHERE /*+ES:
INDEX(dc:title.fulltext,dc:description.fulltext)
OPERATOR(more_like_this) */ ecm:uuid = '1234'
```

will take the most frequent terms of the title and description of document 1234 and find documents that also match those terms. More on JIRA ticket [NXP-25315](#).

Bulk service

The BulkService is a new Nuxeo Platform service that allows to persist a document set homogenous to an NXQL query (and in the future to a page provider) so as to process an "action" on each of the documents. The service allows to get a status on a given "Bulk". It is possible to remotely start a bulk using the [Bulk.RunAction](#) operation that accepts as a parameter the name of the action and an NXQL query for specifying the list of documents on which to run the bulk. A first [Action](#) has been implemented [SetProperty](#) that allows to bulk set a some properties values on a set of documents. Actions can be contributed via an extension point.

More on JIRA tickets [NXP-24837](#), [NXP-25060](#) and [NXP-25097](#).

Annotations Service

Annotation Java Service

A new annotation service has been added, it stores annotations in the repository.

More on JIRA ticket [NXP-24096](#).

Annotation REST API Adapter

The web adapter "annotation" has been added on the document resource so as to retrieve and set annotations on documents using the REST API.

More on JIRA ticket [NXP-24364](#).

Configuration Option for Annotations Location

A new configuration

property `nuxeo.annotations.placeless.storage` is available to change how the annotations are stored. If set to `true` (default), the annotations are placeless documents. If set to `false`, the annotations are stored in a hidden folder. This folder is created under the domain of the annotated document, or under the root if no domain.

More on JIRA ticket [NXP-24863](#).

Comment Service

New Method for Creating an Answer at a Specific Place

- API in `CommentManager` is able to create sub-comments in a specific location
- The `CommentableDocument` adapter is enriched to provide the above service
- The `CommentableDocument` adapter is enriched to provide the service of creating comments in a specific location using

the existing

```
API org.nuxeo.ecm.platform.comment.api.CommentM  
anager#createLocatedComment(org.nuxeo.ecm.core  
.api.DocumentModel,  
org.nuxeo.ecm.core.api.DocumentModel,  
java.lang.String)
```

More on JIRA ticket [NXP-24933](#).

Automation

More Java Objects in Automation Scripting

It's now possible to allow specific Java classes to be used via Automation Scripting, by default we add:

- `java.util.ArrayList`
- `java.util.Arrays`
- `java.util.Collections`
- `java.util.UUID`
- `org.nuxeo.runtime.transaction.TransactionHelpe
r`
- `org.nuxeo.ecm.core.api.Blobs`

- `org.nuxeo.ecm.core.api.impl.blob.StringBlob`
- `org.nuxeo.ecm.core.api.impl.blob.JSONBlob`

Other classes can be added and previously-allowed ones denied, through:

```
<require>org.nuxeo.automation.scripting.classfilter</require>
  <extension
target="org.nuxeo.automation.scripting.internals.AutomationScriptingComponent" point="classFilter">
  <classFilter>
    <allow>com.example.MyClass</allow>
    <allow>com.example.MyOtherClass</allow>
```

```
<deny>org.nuxeo.runtime.transaction.TransactionHelper
</deny>
  </classFilter>
</extension>
```

One can use `<deny>*</deny>` to disallow all previously-allowed classes.

The default contribution now allows scripting code like:

```
function run(input, params) {
  var uuid =
java.util.UUID.randomUUID().toString();
```



```
return  
org.nuxeo.ecm.core.api.Blobs.createJSONBlob("{\"uuid":  
\"\" + uuid + "\"}");  
}
```

More on JIRA ticket [NXP-25020](#).

User Manager

UserGroup.Suggestion Operation Handles Fullname Search

The `SuggestUserEntries` operation performs a full name user search, e.g. typing "John Do" returns the user with first name "John" and last name "Doe". Typing "John" still returns the "John Doe" user and possibly other users such as "John Foo". Respectively, typing "Do" returns the "John Doe" user and possibly other users such as "Jack Donald".

More on JIRA ticket [NXP-24583](#).

Batch upload

Optimised Multipart/Form Upload

When uploading content to Nuxeo using the multi-part/form-data way, no useless copy is made on the way, optimising drastically the upload performance with large videos when using this upload method.

More on JIRA ticket [NXP-24384](#).

User Registration

Stronger Enforcement on Groups Validation for Newly Created Users

Non-administrator users can only invite members from their own group(s).

More on JIRA ticket [NXP-24653](#).

OAuth

Rest API for OAuth 2 Tokens

Some new endpoints have been added to handle CRUD operations on provider and client OAuth tokens.

`GET/PUT/DELETE /oauth2/token/provider/<providerId>/user/<username>` -> perform CRUD on provider tokens

`GET /oauth2/token/provider` -> retrieve all provider tokens for current user

`GET/PUT/DELETE /oauth2/token/client/<clientId>/user/<username>` -> perform CRUD on client tokens

`GET /oauth2/token/client` -> retrieve all client tokens for current user

`GET oauth2/client` -> retrieve all oauth2 clients

`GET` `oauth2/client/<clientId>` -> retrieve an `oauth2 client`

More on JIRA ticket [NXP-24578](#).

Transient Store

Batch Handler

The platform now provides a way to plug custom logics to upload content to a transient store, by contributing a Batch Handler.

```
<extension
target="org.nuxeo.ecm.automation.server.BatchManager"
point="handlers">
<batchHandler>
<name>foo</name>
<class>org.someorg.somepackage.SomeClassThatImplementsBatchHandler</class>
<property name="transientStore">${backingTransientStore}</property>
<property name="key1">value1</property>
<property name="key2">value2</property>
...
<property name="keyN">valueN</property>
</batchHandler>
</extension>
```

An S3 implementation of this batch handler has been added, so as to be able to upload to S3 directly and to benefit from S3 accelerated upload infrastructure (See the new addon here after).

More on JIRA ticket [NXP-24208](#).

Configurable blobProvider for KeyValueBlob transientStore

A `KeyValueBlobTransientStore` can now specify the ids of the key/value store and blob provider to use, instead of defaulting to the name of the transient store itself:

```
<extension
target="org.nuxeo.ecm.core.transientstore.TransientSt
orageComponent" point="store">
<store name="mytransientstore"
class="org.nuxeo.ecm.core.transientstore.keyvalueblob
.KeyValueBlobTransientStore">
<property name="keyValueStore">mykeyvaluestore</
property>
<property name="blobProvider">myblobprovider</
property>
...
</store>
</extension>
```

More on JIRA ticket [NXP-24847](#).

REST API

Nuxeo Rest API Adapters are Now Overridable

`Nuxeo-AllowOverride: true` has been added to the REST module of Nuxeo Platform so that it becomes possible to override an adapter.

More on JIRA ticket [NXP-24532](#).

ManagedBlob Decoder

A generic JSON decoder has been implemented for managed blobs.

More on JIRA ticket [NXP-24925](#).

Key/Value Store: Increment API

New APIs are available on `KeyValueStore`.

Optimized storage of `Long` values:

- `put(String key, Long value)`
- `put(String key, Long value, long ttl)`
- `getLong(String key)`
- `getLongs(Collection keys)`

Atomic increment:

- `addAndGet(String key, long delta)`

More on JIRA ticket [NXP-23745](#).

Packaging / Distribution / Miscellaneous

HSTS Policy

The HSTS header is enabled by default when HTTPS is in use. It forces only HTTPS requests.

More on JIRA ticket [NXP-24254](#).

AWS Client Upgrade

The Nuxeo Platform now uses version 1.11.323 of the Amazon SDK. This notably allows using AWS Comprehend service.

More on JIRA ticket [NXP-24981](#).

JSON output for Log4J

Required dependencies have been added so that it is possible to configure the logs to be serialized as valid JSON. See sample log4j configuration on the linked ticket.

More on JIRA ticket [NXP-25096](#).

JDK Check Enforced in nuxeoctl

A check on the presence of a JDK per is enforced at startup in nuxeoctl

More on JIRA ticket [NXP-21200](#).

Making Use of Tomcat 8 Rewrite Valve

It is now possible to contribute rules to Tomcat Rewrite Valve by leveraging our deployment preprocessor.

Ex (deployment-fragment.xml):

```
<?xml version="1.0"?>
<fragment version="1">
  <!-- JSF permalink redirect -->
  <extension target="rewrite#RULE">
    RewriteRule ^/nxdoc/default/(.*)/view_documents /
ui/#!/doc/$1 [NE,R]
  </extension>

  <!-- ES6 / ES5 code -->
  <extension target="rewrite#RULE">
    RewriteCond %{HTTP_USER_AGENT} .*Chrome.*
    RewriteRule ^/shop/(.*) /shop/es6-bundled/$1 [L]
    RewriteRule ^/shop/(.*) /shop/es5-bundled/$1 [L]
  </extension>
</fragment>
```

More on JIRA ticket [NXP-25040](#).

Multi-Region Replication

A set of addons and scripts have been produced for the ability to replicate all the Nuxeo data in near real time into another hosting region, using Kafka streams for the replication. The intent is to be able to recover from a disaster in less than a minute. Contact Nuxeo for more information about it.

More on JIRA ticket [NXP-24189](#).

Servlet API 3.1

Servlet API 3.1 is now used in Nuxeo code.

More on JIRA ticket [NXP-24386](#).

Addons

Nuxeo Web UI

Orderable Folders

Orderable Folders are now available in Nuxeo Platform with up and down actions. It works with multiple selected documents. For this purpose a new operation is available to order child documents. Navigation tree now takes into account order on Orderable Folders.

More on JIRA ticket [NXP-24254](#).

Trash

Add trash functionality and management to document deletion. Document deletion moves it to trash. In order to manage trash:

- Documents with Folderish facet added a trash pill to manage deleted documents.
- New trash search on the main menu. Has a faceted search on path, size, authors, and text. Trashed documents can be

restore or permanently deleted by users with Manage Everything permission. A new [EmptyTrash](#) operation allows to permanently delete a Folderish trash content which is available on the Folderish trash pill UI. Finally, a set of functional tests for new trash features.

More on JIRA ticket [NXP-23798](#).

Direct Upload to 3rd-Party Service

Batch upload refactored to support third party providers. It is possible to integrate providers for feature rich and performance upload. To this end, the upload behaviour now supports external providers and allows features like progress and multipart.

More on JIRA ticket [NXP-24269](#).

User Invitation Link

Fixed link on user invitation e-mail that led to "page not found".

More on JIRA ticket [NXP-24535](#).

Remove from Collection

The "Remove from Collection" action is now displayed on all document types with the Collection facet.

More on JIRA ticket [NXP-24347](#).

Close Drawer Action

A new close action button has been added to let you hide the drawer. It appears on the middle right side of the drawer.

More on JIRA ticket [NXP-24082](#).

New Languages

Italian, Dutch and Swedish languages have been added to Web UI and Nuxeo Elements.

More on JIRA tickets [NXP-24451](#) and [NXP-24445](#).

User Cloud Settings

Adds an interface to allow users to manage their OAuth2 tokens. Managing inbound and outbound authorisations. [NXP-20773](#) The user has a way to manage all the cloud services permission tokens granted to the Nuxeo account. [NXP-22588](#) Users can now manage external applications authorisations with Nuxeo. [NXP-24841](#) Improves performance on Cloud Service area to lazy load pages. [NXP-24578](#) New Rest API for OAuth 2 tokens.

Some new endpoints have been added to handle CRUD operations on provider and client OAuth tokens:

```
GET/PUT/DELETE /oauth2/token/provider/<providerId>/user/<username> -> perform CRUD on provider tokens
```

```
GET /oauth2/token/provider -> retrieve all provider tokens for current user
```

```
GET/PUT/DELETE /oauth2/token/client/<clientId>/user/<username> -> perform CRUD on client tokens
```

`GET /oauth2/token/client` -> retrieve all `client` tokens for current user

`GET /oauth2/client` -> retrieve all oauth2 clients

`GET /oauth2/client/<clientId>` -> retrieve an oauth2 client

More on JIRA ticket [NXP-24252](#)

Drag and drop

It is now possible to use drag and drop user interaction to move and copy documents to folders and collections on a listing.

[NXP-24351](#) Enables user to add documents to sibling collections by drag and drop. [NXP-24350](#) Enables user to move documents to sibling folders by drag and drop.

More on JIRA ticket [NXP-22807](#)

Document metadata differences

Documents can be compared to other documents or within its versions. Common schemas are compared. UI is fully responsive and can show only different data or all of it. The comparison UI is fully configurable.

[NXP-24780](#) Allows documents to be selected to compare. [NXP-24781](#) Permits user to pick a different version to be compared. [NXP-24786](#) Enables switching comparing documents position. [NXP-24784](#) Visualizes metadata differences on two documents. [NXP-24784](#) Allows custom element to be added to compare different metadata properties. Defines custom elements to be used on blobs, document and user references.

Default comparison elements

Allows custom element to be added to compare different metadata properties. Defines custom elements to be used on blobs, document and user references.

Blobs: In case of blobs, the name should be the content for comparison. An action should also be present so that if the MD5 of those blobs is different, the action allows the user to launch a blob content comparison (using ARender comparison).

Document references: should compare the document title

User references: should compare with the user tag widget

A custom element was introduced to compare documents: `nuxeo-diff`. This element now dynamically loads the file `diff/imports.html`, which imports custom elements from the `diff/elements` folder and registers them in a global registry using a set of rules. These rules define for what property names or property types should this elements be used to represent the differences (or the raw value in case there are no differences). A sample of a valid `diff/imports.html` follows:

```
<link rel="import" href="elements/nuxeo-default-diff.html">
<link rel="import" href="elements/nuxeo-blob-diff.html">
<link rel="import" href="elements/nuxeo-dcdescription-diff.html">
<script>
```

```
Nuxeo.Diff.registerElement('nuxeo-blob-diff',  
{type: 'blob'});  
Nuxeo.Diff.registerElement('nuxeo-dcdescription-  
diff', {property: 'dc:description'});  
</script>
```

These elements are used as children by `nuxeo-object-diff`, which is the element responsible for representing a diff for a given property, loading those elements based on the property name or property type. In case no custom element is registered for the given criteria, the default diff element (named `nuxeo-default-diff`) is used.

In order to allow the current custom elements to be overridden and new elements to be added, the following files are not vulcanized:

- `nuxeo-diff-page.html`
- `diff/imports.html`
- `diff/elements/.html`

The `nuxeo-diff-page` stamps `nuxeo-diff`, which can be parameterized. `nuxeo-diff` exposes several customization properties, including the headers, enrichers and schemas used to fetch the documents. The headers property can be used to determine whether documents are fetched with or without resolved entities.

More on JIRA ticket [NXP-24785](#)

User Cloud Settings

Introduces new file-based document pill to allow annotations with Arender.

More on JIRA ticket [NXP-25107](#)

New Languages

Adds Chinese simplified and Hebrew locales for Web UI.

More on JIRA ticket [NXP-25112](#).

Performance improvements

Several actions to provide better performance on Web UI.

[NXP-25158](#) Enables a way to skip aggregates computation, making the request potentially faster. This is used on a range or pagination queries as there is no need to update aggregates. This can be used by adding `skipAggregates=true` as a HTTP header when invoking the search rest endpoint. [NXP-25202](#) Adds an initial progress page for fast feedback on loading. [NXP-25139](#) Updates cache headers for Web UI static resources. [NXP-24820](#) The memory footprint of the WebUI has been reduced. [NXP-24422](#) Edge performance was analysed and new improvement actions were planned.

More on JIRA ticket [NXP-25038](#)

UX improvements

On 10.2, several UX improvements were added to Web UI:

[NXP-25021](#) A login banner is displayed when the WebUI session has expired. [NXP-25202](#) Adds an initial progress page for fast feedback on loading. [NXP-24648](#) Improves UX by notifying user when connection is offline making Web UI not reacting. [NXP-24752](#) Introduces mixin that allows selection toolbar appearance and position customisation. Adds theme variable for selection color and uses this by default. [NXP-24692](#) To assure the best legibility, the color of the "shortcut" icons was changed to have a high contrast to the background color. [NXP-22062](#) Document UIDs on audit log now have a link back to document. [NXP-24894](#) Fixes drawer queue keyboard navigation. [NXP-24466](#) Document information included in administration audit logs.

Bugfixes

On 10.2, several relevante bugfixes were included to Web UI:

[NXP-25310](#) Groups with slash in their name are supported. [NXP-24789](#) The access to the user administration is disabled for anonymous users. [NXP-25094](#) Makes bulk download notification fixed with no intermittency. [NXP-24832](#) Ordered folder support is extended to document types which inherit from OrderedFolder type. [NXP-22555](#) `<nuxeo-document-distribution-chart>` not restricted to default-domain anymore..

Nuxeo Elements

Migration to ES6

Migrates all elements to ES6 class-based syntax which allows usage of Polymer 2 and move away from Polymer legacy APIs

[ELEMENTS-506](#) Migrates nuxeo-dataviz-elements to ES6. [ELEMENTS-505](#) Migrates nuxeo-elements to ES6. [ELEMENTS-640](#) Migrates Widgets to ES6 and Polymer 2. [ELEMENTS-635](#) Migrates UI Elements behaviors, slots and filters to ES6 and Polymer 2. [ELEMENTS-637](#) Migrates Actions and Viewers to ES6 and Polymer 2. [ELEMENTS-639](#) Migrates Listing elements to ES6 and Polymer 2. [ELEMENTS-636](#) Migrates Document Permissions and User Management to ES6 and Polymer 2. [ELEMENTS-638](#) Migrates Demos and Tests to ES6 and Polymer 2.

More on JIRA ticket [ELEMENTS-504](#).

Performance improvements

Several actions to provide better performance on Nuxeo Elements.

[ELEMENTS-673](#) Improves performance on lazy loading to not fetch already loaded documents [ELEMENTS-701](#) Adds option to skip aggregation computation in nuxeo-page-provider for better performance.

More on JIRA ticket [NXP-25038](#)

UX improvements

Several UX improvements were added to Nuxeo Elements that impact Web UI 10.2:

[ELEMENTS-685](#) Removes background on image previewer and exposes it as an attribute for better visual. [ELEMENTS-707](#) Improves requirement symbol on form's labels. [ELEMENTS-666](#) Improves nuxeo-user-suggestion UI to make it non-ambiguous. [ELEMENTS-622](#) Selection UI improved to be consistent in all view mode. [ELEMENTS-687](#) Changes image previewer dragging behaviour to contain image on bounds. [ELEMENTS-677](#) Changes blob delete action to distinguish from document trash.

Update Nuxeo JS client dependency

Adapts Element's tests for nuxeo js client 3.6.1

More on JIRA ticket [ELEMENTS-650](#)

Sort aggregates elements

Aggregation widgets can now be sorted by the label with the *sort-by-label* boolean property. i.e. `<nuxeo-checkbox-aggregation sort-by-label .../>`

More on JIRA ticket [ELEMENTS-621](#)

Elements improvements and additions

[ELEMENTS-669](#) Introduces new element . [ELEMENTS-662](#) Changes `nuxeo-date-picker` element to use vaadin date picker for better browser support. [ELEMENTS-662](#) Two new elements, `<nuxeo-directory-checkbox>` (multiple) and `<nuxeo-directory-`

[radio-group](#)> (single) added to present and select directory entry(ies). [ELEMENTS-624](#) Exposes [rows](#) attribute in nuxeo-textarea element. [ELEMENTS-610](#) Replaces video javascript library with browser native video element for to better support browsers.

Amazon S3 Direct Upload for Web UI

New add-on to upload using AWS S3 infrastructure with support for multipart. Allows future integration of other providers. Integrated with Web UI upload with real time upload progress. For 10.2 we have added AWS instance role support for better security.

More on JIRA ticket [NXP-24490](#) and [NXP-24748](#).

ARender Connector and Document, Image, Video Annotations

A first implementation of the ARender SPI bridge has been done so as to be able to preview content stored in Nuxeo using the [ARender previewer](#). It allows to preview and annotate content, may it be an office file, an image or a video. A first integration to Web UI is done in the addon, the ARender previewer appears in a new "annotations" tab. Deeper integration will be done in the future. Also, in 10.3, comments made for a given annotation will be synced with Nuxeo Comments. A package is available on the marketplace.

More on JIRA ticket [NXP-24104](#).

Nuxeo Platform Importer

Image and Video capabilities for the Nuxeo Stream Random Importer

The Nuxeo importer has an option for filling the repository with random content. This content can now be images or videos, randomly produced by watermarking a set of sample media.

More on JIRA ticket [NXP-24318](#).

IMAP Connector

Nuxeo IMAP Connector addon is now available on Web UI. You can create and configure IMAP folder documents on WebUI, with a sync action to import all unread emails from account. This addon adds a new custom list view for emails on email folder documents and a custom view layout to email message documents with relevant information about the content, senders, receivers and attachments.

More on JIRA ticket [NXP-23951](#).

Drive (Server part)

The release notes of the Drive client part can be found on [GitHub](#).

.lnk Files Ignored

Windows symlink files [.lnk](#) are now ignored by default

More on JIRA ticket [NXP-24490](#).

Full Scan Query Optimized

A great optimisation has been added lowering heavily the charge of the Elasticsearch cluster when using Nuxeo Drive with Nuxeo.

More on JIRA ticket [NXP-24232](#) and [NXP-24637](#)

Nuxo DAM

Improve Video Processing

The video info (duration, format, etc.) is now computed by an asynchronous work to avoid loading the blob and running `ffmpeg-info` synchronously. This work, in turn, schedules two asynchronous works to process the video storyboard and conversions.

As a consequence, the user might not have the video info in the UI immediately after creating / updating a Video document, needing to refresh the page once the asynchronous work is done. This change allows a better behaviour when bulk importing videos.

More on JIRA ticket [NXP-24316](#).

Nuxeo Vision

All Backend Service features can be leveraged in Nuxeo Vision

The features to use (and sent to the provider) are no more checked against a predefined list. This allows to use any new feature available without waiting for an update of the plugin. After using

such new feature, the caller should use the `getNativeObject` method and handle the results based on the provider's documentation.

More on JIRA ticket [NXP-24499](#).

Nuxeo Quota

Operation to Recompute Partially Quotas

A new operation: `Quotas.RecomputeStatistics` is provided, with optional parameters:

- `tenantId / username / path` (only one allowed)
- `updaterName` (defaults to `documentsSizeUpdater`)

More on JIRA ticket [NXP-21017](#).

Farewell

WebEngine GWT Integration

The WebEngine GWT Integration that used for the former annotations system has been removed.

More on JIRA ticket [NXP-24317](#).