Why talking about Renditions ?

There are several contextual reasons :

- we have a 5.5 addon nuxeo-platform-rendition that allow to publish PDF renditions of a document
- this would make sense to have the new addon rendering-template being renamed rendition-template
- exposing renditions like PDF convertion or HTML preview would make sense at least in the context of CMIS

Current Rendition service

Current use case

The current use case is simple : *"when I publish a File document with attached Word file, I would like it to be published as a Nuxeo Document + PDF conversion of the Word file"*.

The use case was the reason why the 5.5 addon was created.

Current architecture

Currently inside the Rendition Addon we have :

RenditionOperation

Operation or chain used to generate the rendition blob. - inputs : Blob - output : Blob

RenditionDefinition

XMap Object used to register rendition, it contains - name of the rendition - RenditionOperation name

Rendition document

This is a copy of the versionned Document (same doc type) where

- the main Blob is replaced by the output of the RenditionOperation
- a facet is added to link it to the version uuid + live uuid

RenditionCreator

This is an unrestricted runner that :

- create a version of the target document
- create an unfilled copy of the version
- update the main Blob from the output of the RenditionOperation
- set an ACL so that current user can see it (means that by default a rendition Doc is only visible by the user that created it)

RenditionService

NXRuntime component and service :

- manage RenditionDefinition registration
- provides API to create a rendition (encapsulate RenditionCreator)

Publisher plugin

This plugin allows to publish a rendition of the source Document

- create the rendition
- publish a proxy to the rendition

Extending Rendition concept

rendition-template

Current status

The current module "rendering template" addon provides :

- a way to create a Template Document from an ODT/Docx file with merge fields
- a way to associate a Template Document to a Document (manual association)
- a way to associate a Template Document to a Document type (automating binding)

The Template Document is used to render the associated document. The resulting output can be :

- native template format (ODT, docX)
- result of a conversion (ODT, docX, PDF ...)

The direct use cases are :

"I create a Word document using my company template so that some parts of the document (headers, coverpage, ...) are normalized and always up to date with the Nuxeo Document's attribute (author,

modification date ...)"

"I create a Note document and I want to render it with a converpage, an index, an history page."

"I want to have a custom editable view of my document, like VIF product doc."

At least for the 2 first use case, it would make sense to generate a rendition at publishing time.

Possible extensions

The current addon has a pluggable templateprocessing engine that for now supports :

- XDocReport
- JODReport (that will be deprecated soon)
- raw XML docx processor (legacy of TWC prototype)

It's very easy to add : - a bare Freemarker processor that allow to do HTML rendition - (an iText processor to manage PDF custom views)

One of the current limitation is that you can associate only one template with a Document => you have only one templace based rendition ==> we should probably allow to have several ?

Changes needed in Rendition model

In order to do the alignment, there are several changes that need to be made in the current Rendition service.

RenditionOperation

The input should be a DocumentModel and not only a Blob.

NB : *We may also want to automatically provide additional contextual information like parameters.*

RenditionDefinition

In addition of the existing fields I think we should add :

- a flag to indicate if Rendition should be stored or not

- a filter (like action filter) that allows to determine if the contributed Rendition is actually available on a given DocumentModel

About the storage Flag, the idea is that is some cases, it make sense to be able to list and execute renditions without actually have to store them.

As simple examples :

- PDF rendering could be seen as an "on the fly" unstored rendition

- HTML preview could be seen as an "on the fly" unstored rendition

=> this is also a way to have a solution to give access to all conversions

In addition, since the template binding can be done on a per-instance basis, and association is stored as content this does not match with a static extension point registration. => when I want to list available renditions on a document, I can not only list thee registry

==> need a way to contribute a RenditionProvider impl ?

Rendition document

So far I don't have a better idea than what exists now. The fact to store the full snapshot of the Document as well as the Blob rendition does make sense when we want to store the result.

The only point could be that if I apply the same model to a Note document, I won't be able to store the rendition Blob as main Blob of the Note doc.

=> need to store it in a dedciated Blob schema ????

In the case of "on the fly" Rendition the output would be a blob and no Rendition Document get's created.

Suggestion of Alain: In some cases, it makes sens to let the template as the blob of the checkout document, and to let the renditions as the main blob of all the versions made from it.

RenditionCreator

May be we could make this class pluggable via RenditionDefinition, but I don't see direct use cases.