Nuxeo 5.4 : les nouveautés

Atelier GED - 30 mars 2011, Paris

Nuxeo 5.4 : les nouveautés

- ## Nuxeo.conf et templates

Depuis la version 5.3.2, nouvelle façon de configurer Nuxeo à l'aide du fichier nuxeo.conf et des templates.

Les templates contiennent (notamment) les fichiers de configurations qui seront copiés dans le répertoire config du serveur (nxserver).

Il existe plusieurs templates : common, default, mysql, mssql, oracle, postgresql et custom.

Principe :

- Recherche du (ou des) templates choisis dans nuxeo.conf.
- Pour chaque template, lecture du fichier nuxeo.defaults pour définir de nouvelles valeurs par défaut (attention à l'ordre d'inclusion des templates).
- Les variables non renseignées précédemment sont ensuite initialisées avec le fichier templates/nuxeo.defaults.
- Enfin, le fichier nuxeo.conf permet de remplacer les valeurs par défaut par ses propres paramètres.

Nuxeo 5.4 : les nouveautés

Le template **custom** :
Ce template permet de centraliser dans un même répertoire des fichiers permettant de personnaliser la configuration de Nuxeo.

- Dans nuxeo.conf, nuxeo.templates=custom

- Dans templates/custom/nuxeo.defaults :
renseigner nuxeo.template.includes avec les templates que l'on souhaite inclure
renseigner custom.target (là où seront copiés les fichiers contenus dans custom)

- Le fichier **nuxeo.conf** : permet notamment de définir les répertoires où sont stockés les données binaires (data), les logs, les ports, les paramètres de connexion à la base de données … Liste exhaustive des paramètres : https://doc.nuxeo.com/display/NXDOC/Configuration+parameters+index

L'Admin Center permet d'éditer le fichier nuxeo.conf depuis l'interface de nuxeo. Depuis la version 5.4.1, un wizard, lancé par défaut lors de la première l'installation permet également de configurer ces paramètres.

Nuxeo 5.4 : les nouveautés

Tomcat

Depuis la version 5.4, nuxeo supporte aussi bien l'utilisation de Tomcat que de Jboss (en production). Il est même conseillé de privilégier Tomcat (ce qui est plutôt une bonne nouvelle) => performances supérieures.

Nouveau Shell (le précédent s'appuyait sur RMI et n'était donc disponible que pour Jboss) qui s'appuie notamment sur le client content automation.

Admin Center

Interface permettant :
de visualiser/modifier les paramètres de nuxeo.conf, de visualiser les paquets déployés ...
de consulter quelques statistiques (nb de documents, d'espaces, plus gros fichiers …)
d'utiliser quelques outils de monitoring.
d'accéder à Nuxeo Market Place (téléchargements et installations automatiques des mises à jour et des add-ons => compte connect nécessaire …)
de paramétrer des services openSocial

Nuxeo 5.4 : les nouveautés

NXQL

Extension des opérations de recherche basées sur nxql, notamment :

- support de ILIKE

- recherche plein-texte acceptant l'utilisation du terme OR, des négations et des guillemets (expression exacte)

Versioning

Nuxeo a changé sa gestion des versions d'un document en introduisant les notions de Check In et Check Out. Ces opérations sont quasiment transparentes pour l'utilisateur depuis l'interface mais sont utilisées par les API de bas niveau.

Check In : opération permettant de créer une version d'un document de travail (working Copy). Ce dernier n'est alors plus modifiable.

Check Out : opération permettant de rendre disponible (pour modification) un document.

Ces changements ont été notamment dictés pour une meilleure compatibilité avec CMIS.

Nuxeo 5.4 : les nouveautés

CMIS

Amélioration du support CMIS (OpenCMIS d'Apache Chemistry project). Correction des bugs de la 5.3, gestion des versions, meilleur support des requêtes CMISQL (recherche full text) …

Editeur de thèmes

Nouvel éditeur, permettant une personnalisation assez poussée (éditeur de CSS, éditeur de cannevas, gestion des fragments ...).

Recherches par filtres (faceted search)

Il s'agit d'une nouvelle forme de navigation, complètement paramétrable (via les content views), qui permet à l'utilisateur d'enregistrer des recherches et d'en lister les résultats régulièrement. Par défaut les recherches sont stockées dans un répertoire saved_searches dans l'espace personnel de l'utilisateur.

Nuxeo 5.4 : les nouveautés

Personnalisation : content views, layout et widgets

Les content-views :

Ils permettent de définir des listes de documents : on définit les critères d'obtention de ces documents et la façon dont on souhaite qu'ils s'affichent.

Exemples : la liste des documents contenus dans un espace de travail s'appuie sur ce mécanisme, les recherches par filtres aussi.

```
<extension target="org.nuxeo.ecm.platform.ui.web.ContentViewService"
  point="contentViews">
  <contentView name="document_content">
    <coreQueryPageProvider>
     <property name="coreSession">#{documentManager}</property>
```

Nuxeo 5.4 : les nouveautés

```
<!-- La requete NXQL suivi des paramètres (ici un seul) qui remplaceront les ? (attention à l'ordre), ici on sélectionne tous les documents enfants d'un document donné (paramètre) s'il n'est pas de type hidden et n'a pas été supprimé, on précise un ordre de tri et un nombre de résultat par défaut  -->
    <pattern>
      SELECT * FROM Document WHERE ecm:parentId = ?
      AND ecm:mixinType != 'HiddenInNavigation'
      AND ecm:currentLifeCycleState != 'deleted'
    </pattern>
    <parameter>#{currentDocument.id}</parameter>
    <sort column="dc:title" ascending="true" />
    <pageSize>20</pageSize>
  </coreQueryPageProvider>
```

Nuxeo 5.4 : les nouveautés

Affichage des résultats :

```
<resultLayouts>
    <layout name="document_listing_ajax" title="document_listing" translateTitle="true"
iconPath="/icons/document_listing_icon.png" showCSVExport="true" showPDFExport="true" />
    <layout name="document_listing_ajax_compact_2_columns"
title="document_listing_compact_2_columns"  translateTitle="true"
iconPath="/icons/document_listing_compact_2_columns_icon.png" />
    <layout name="document_listing_ajax_icon_2_columns"  title="document_listing_icon_2_columns"
     translateTitle="true" iconPath="/icons/document_listing_icon_2_columns_icon.png" />
  </resultLayouts>
  <selectionList>CURRENT_SELECTION</selectionList>
  <actions category="CURRENT_SELECTION_LIST" />
 </contentView>
</extension>
```

Nuxeo 5.4 : les nouveautés

Associer un type de document à un content view (exemple ecm-types-contrib.xml):

```
<type id="Workspace">
    <label>Workspace</label>
    <icon>/icons/workspace.gif</icon>
    <default-view>view_documents</default-view>
    <create-view>create_workspace</create-view>
    ...
    <contentViews category="content">
      <contentView>document_content</contentView>
    </contentViews>
    <contentViews category="trash_content">
      <contentView showInExportView="false">
        document_trash_content
      </contentView>
    </contentViews>
  </type>
```

Nuxeo 5.4 : les nouveautés

Le rendu de tout ça :

Le fichier qui gère l'onglet contenu (document_content.xhtml, défini dans actions-contrib.xml) teste s'il existe un content view associé au type de document, si oui, il va chercher document_content_view.xhtml qui lui même appelle content_view.xhtml qui va (entre autres) afficher les résultats en utilisant le layout défini dans le content view.

Layouts et widgets

Un layout est un fichier xml servant essentiellement à l'affichage des métadonnées liées à un document. Il est constitué d'un assemblage de widgets qu'il définit et dont il gère le rendu et l'ordonnancement.

Notions de mode aussi bien pour les widgets (edit, view) que les layouts (edit, view, create, summary ...).

Nuxeo 5.4 : les nouveautés

Widget et layout se définissent dans des point d'extension :

<extension target="org.nuxeo.ecm.platform.forms.layout.WebLayoutManager"

point="layouts"> ou point="widget"

Petit exemple simple :

```
<layout name="heading">
    <templates>
      <template mode="any">/layouts/layout_default_template.xhtml</template>
    </templates>
    <rows>
      <row><widget>title</widget></row>
      <row><widget>description</widget></row>
    </rows>
    <widget name="title" type="text">
      <labels>
        <label mode="any">label.dublincore.title</label>
      </labels>
      <translated>true</translated>
      <fields>
```

Nuxeo 5.4 : les nouveautés

```
        <field>dc:title</field>
      </fields>
      <properties widgetMode="edit">
        <property name="required">true</property>
      </properties>
    </widget>
    <widget name="description" type="textarea">
      <labels>
        <label mode="any">label.dublincore.description</label>
      </labels>
      <translated>true</translated>
      <fields>
        <field>dc:description</field>
      </fields>
    </widget>
  </layout>
</extension>
</component>
```

Nuxeo 5.4 : les nouveautés

Types de widget : Text, Int, DateTime, Textarea, Secret, Template, File, SelectOneDirectory, SelectMultipleDirectory, htmlText, list, checkbox

Utilisation des layourts : tags JSF disponibles :

```
<div xmlns="http://www.w3.org/1999/xhtml"
    xmlns:nxl="http://nuxeo.org/nxforms/layout">
  <nxl:documentLayout mode="view" value="#{currentDocument}" />
</div>
```

= > permet d'afficher le layout associé au document courant

```
<div xmlns="http://www.w3.org/1999/xhtml"
    xmlns:nxl="http://nuxeo.org/nxforms/layout">
  <nxl:layout name="heading" mode="view" value="#{currentDocument}" />
</div>
```

=> force l'utilisation d'un layout sur le document courant même si celui-ci n'a pas été défini dans le type du document.