

Personnaliser le thème

- [Principe de fonctionnement](#)
- [Application concrète](#)



La gestion des thèmes a quelque peu été modifiée avec la version 5.5 de Nuxeo.
La documentation ci-dessous s'applique donc dans un environnement Nuxeo 5.5

Principe de fonctionnement

Les thèmes sont constitués de trois éléments principaux :

- Une structure (en xml) qui définit les en-tête, pied de page, menu de gauche, cadre principal : le layout
- Des css
- Une nouvelle notion : les flavors qui permettent de définir les couleurs, le logo, les polices ... On peut ainsi, si on le souhaite facilement changer l'apparence de Nuxeo suivant le context

Le layout est donc un fichier xml qui définit la structure d'une page et les éléments qui sont appelés (widgets) dans chaque zone.

Ce dernier doit être référencé dans un point d'extension spécifique :

```
<extension target="org.nuxeo.theme.services.ThemeService" point="themes">
  <theme>
    <src>themes/esup-document-management.xml</src>
  </theme>
</extension>
```

Voici un exemple du contenu du fichier esup-document-management.xml (à noter que depuis la version 5.5 de Nuxeo, il est possible d'ajouter des classes sur les attributs <page>, <section>, <cell> et <fragment>, ces derniers pouvant ensuite être définies dans un css)

```
<?xml version="1.0" encoding="UTF-8"?>
<theme name="esup" template-engines="jsf-facelets">
  <layout>
    <page name="default" class="documentManagement">
      <section class="nxHeader">
        <cell class="logo">
          <!-- logo -->
          <fragment type="generic fragment"/>
        </cell>
        <cell class="content">
          <!-- header -->
          <fragment type="generic fragment"/>
        </cell>
        <cell class="includes">
          <!-- Include -->
          <fragment type="generic fragment"/>
        </cell>
      </section>
      <section class="nxMainContainer">
        <cell class="column">
          <!-- tree view -->
          <fragment perspectives="default,multiple_domains" type="generic fragment"/>
          <!-- clip board -->
          <fragment perspectives="default,multiple_domains" type="generic fragment"/>
          <!-- calendar -->
          <fragment perspectives="view_calendar" type="generic fragment"/>
        </cell>
        <cell class="content">
          <!-- breadcrumbs -->
          <fragment type="generic fragment" class="breadcrumb"/>
          <!-- main area -->
          <fragment type="region fragment" class="mainFragment"/>
        </cell>
      </section>
      <section class="nxFooter">
```

```

        <cell class="content">
            <!-- copyrights -->
            <fragment type="generic fragment"/>
        </cell>
    </section>
</page>
<page name="popup">
    <section class="nxMainContainer">
        <cell class="content">
            <!-- Include -->
            <fragment type="generic fragment"/>
            <!-- main area -->
            <fragment type="region fragment"/>
        </cell>
    </section>
</page>
<page name="print">
    <section class="nxMainContainer">
        <cell class="content">
            <!-- includes -->
            <fragment type="generic fragment"/>
            <!-- breadcrumbs -->
            <fragment type="generic fragment"/>
            <!-- main area -->
            <fragment type="region fragment"/>
        </cell>
    </section>
</page>
</layout>
<properties element="page[1]/section[2]/cell[2]/fragment[2]">
    <name>body</name>
    <defaultBody/>
    <defaultSrc/>
</properties>
<properties element="page[2]/section[1]/cell[1]/fragment[2]">
    <name>body</name>
    <defaultBody/>
    <defaultSrc/>
</properties>
<properties element="page[3]/section[1]/cell[1]/fragment[3]">
    <name>body</name>
    <defaultBody/>
    <defaultSrc/>
</properties>
<formats>
    <widget element="">
        <view>theme view</view>
        <icon>${org.nuxeo.ecm.contextPath}/icons/favicon.png</icon>
        <charset>utf-8</charset>
    </widget>
    <widget element="page[1]|page[2]|page[3]">
        <view>page frame</view>
    </widget>
    <widget element="page[1]/section[1]|page[1]/section[2]|page[1]/section[3]|page[2]/section[1]|page[3]/section
[1]">
        <view>section frame</view>
    </widget>
    <widget element="page[1]/section[1]/cell[1]|page[1]/section[1]/cell[2]|page[1]/section[1]/cell[3]|page[1]
/section[3]/cell[1]|page[2]/section[1]/cell[1]|page[3]/section[1]/cell[1]">
        <view>cell frame</view>
    </widget>
    <widget element="page[1]/section[1]/cell[1]/fragment[1]">
        <view>Esup logo</view>
    </widget>
    <widget element="page[1]/section[1]/cell[2]/fragment[1]">
        <view>Esup header</view>
    </widget>
    <widget element="page[1]/section[1]/cell[3]/fragment[1]">
        <view>nuxeo5 includes</view>
    </widget>
    <widget element="page[1]/section[2]/cell[1]">

```

```

    <view>cell frame</view>
</widget>
<widget element="page[1]/section[2]/cell[2]">
    <view>cell frame</view>
</widget>
<widget element="page[1]/section[2]/cell[1]/fragment[1]">
    <view>nuxeo5 tree explorer</view>
</widget>
<widget element="page[1]/section[2]/cell[1]/fragment[2]">
    <view>nuxeo5 clip board</view>
</widget>
<widget element="page[1]/section[2]/cell[2]/fragment[1]">
    <view>nuxeo5 breadcrumbs</view>
</widget>
<widget element="page[1]/section[2]/cell[2]/fragment[2]|page[2]/section[1]/cell[1]/fragment[2]">
    <view>facelet region</view>
</widget>
<widget element="page[1]/section[3]/cell[1]/fragment[1]">
    <view>Esup footer</view>
</widget>
<widget element="page[2]/section[1]/cell[1]/fragment[1]">
    <view>nuxeo5 includes</view>
</widget>
<!-- Page 3 : print page -->
<widget element="page[3]/section[1]/cell[1]/fragment[1]">
    <view>nuxeo5 includes</view>
</widget>
<widget element="page[3]/section[1]/cell[1]/fragment[2]">
    <view>nuxeo5 breadcrumbs</view>
</widget>
<widget element="page[3]/section[1]/cell[1]/fragment[3]">
    <view>facelet region</view>
</widget>
</formats>
</theme>

```

Les vues que nous avons déclarées dans ce fichier doivent également l'être dans un point d'extension spécifique.

Si nous reprenons l'exemple de la vue Esup logo, cette dernière doit être déclarée comme suit :

```

<extension point="views" target="org.nuxeo.theme.services.ThemeService">
    <view name="Esup logo" template-engine="jsf-facelets">
        <format-type>widget</format-type>
        <template>incl/esup_logo.xhtml</template>
    </view></extension>

```

Les vues sont associées à des templates xhtml qu'il faudra eux aussi créer.

Toujours pour illustrer notre exemple, voici le contenu du fichier esup_logo.xhtml :

```

<div xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:nxu="http://nuxeo.org/nxweb/util"
    xmlns:c="http://java.sun.com/jstl/core" class="menu">
<div class="siteLogo">
    <h:form>
        <h:commandLink action="#{navigationContext.goHome()}">
            <nxu:set var="flavorLogo" value="#{themeActions.logo}"
                cache="true">
                <c:if test="#{empty flavorLogo}">
                    <h:graphicImage
                        value="/img/esup_logo.jpg"
                        alt="Esup"
                        title="Esup"
                        width="100"
                        height="37" />
                </c:if>
                <c:if test="#{not empty flavorLogo}">
                    <h:graphicImage
                        value="#{flavorLogo.path}"
                        alt="#{flavorLogo.title}"
                        title="#{flavorLogo.title}"
                        width="#{flavorLogo.width}"
                        height="#{flavorLogo.height}" />
                </c:if>
            </nxu:set>
        </h:commandLink>
    </h:form>
</div>
</div>

```

Nous y définissons une image avec des caractéristiques par défaut au cas où la variable flavorLogo ne soit pas renseignée. Dans le cas contraire, ce sont les données relative à cette dernière qui sont prises compte.

La définition de flavor se fait aussi dans un point d'extension : voici comment nous avons déclaré le flavor defaultEsup

```

<extension point="flavors" target="org.nuxeo.theme.styling.service">

  <flavor name="defaultEsup">
    <label>label.theme.flavor.nuxeo.default</label>
    <!-- donnees relative au logo -->
    <logo>
      <path>/img/logoBx1.png</path>
      <previewPath>/img/logoBx1.png</previewPath>
      <width>269</width>
      <height>45</height>
      <title>Bordeaux 1</title>
    </logo>
    <!-- Definition des fichiers dans lesquels on va notamment definir les couleurs et polices principales -->
    <presetsList>
      <presets category="border" src="themes/palettes/default-borders.properties"/>
      <presets category="background" src="themes/palettes/default-backgrounds.properties"/>
      <presets category="font" src="themes/palettes/default-fonts.properties"/>
      <presets category="color" src="themes/palettes/default-colors.properties"/>
    </presetsList>
    <!-- previsualisation de la palette de couleurs, utilise si on permet le changement de theme par
workspace par exemple -->
    <palettePreview>
      <colors>
        <color>#cfecff</color>
        <color>#70bbff</color>
        <color>#4e9ae1</color>
        <color>#fff</color>
        <color>#fff</color>
        <color>#0079a6</color>
        <color>#00416b</color>
        <color>#332e2e</color>
        <color>#cfecff</color>
      </colors>
    </palettePreview>
  </flavor>
</extension>

```

Les presets peuvent également être utilisés dans les fichiers css : par exemple si on a

```

.nxHeader {
  background-color: "header (__FLAVOR__ background)";
  overflow: auto; width:100% ;
}

```

on récupère la couleur définie pour header dans le fichier themes/palettes/default-backgrounds.properties (category=background).

themes/palettes/default-backgrounds.properties :

```

color.major.light=#e6f1ff
color.major.medium=#cfecff
color.major.strong=#adf
color.major.stronger=#4e9ae1
neutral.lighter=#f6f6f6
neutral.light=#f1f2f3
neutral.medium=#eeeff0
neutral.strong=#dfe0e0
neutral.stronger=#8b8b8b
button=#e7e7e7
button.disabled=#f6f6f6
tabs.selected=#fff
subtabs.selected=#fff
info=fff3cf
warning=ffc3a6
header=#486dbc
footer=#2a3f68
breadcrumbs=#ecc7a6

```

Pour pouvoir utiliser notre thème, il nous manque encore quelques point d'extension qu'il nous faut ajouter : Pages, styles et applications.

Pages :

```

<extension target="org.nuxeo.theme.styling.service" point="pages">

  <themePage name="esup/default">
    <!-- on utilise le flavor cree plus haut -->
    <defaultFlavor>defaultEsup</defaultFlavor>
    <flavors>
      <flavor>default</flavor>
      <flavor>rainbow</flavor>
    </flavors>
    <!-- on insere les css qui sont prises en compte dans cet ordre
    Pour etre certain que notre css personnalisee soit prise ne compte, on la declare donc en dernier-->

    <styles>
      <style>basics</style>
      <style>buttons_and_actions</style>
      <style>header</style>
      <style>body</style>
      <style>footer</style>
      <style>navigation</style>
      <style>tables</style>
      <style>forms</style>
      <style>popups</style>
      <style>breadcrumb</style>
      <style>messages_and_tooltips</style>
      <style>foldable_boxes</style>
      <style>drag_and_drop</style>
      <style>richfaces_components</style>
      <style>tabs_content</style>
      <style>nuxeo_dm_specific</style>
      <style>esupStyle</style>
    </styles>
  </themePage>
</extension>

```

Nous avons déclaré l'utilisation d'une css esupStyle, il faut donc la définir dans un point d'extension **styles** :

```
<extension target="org.nuxeo.theme.styling.service" point="styles">
  <style name="esupStyle">
    <src>themes/css/esupStyle.css</src>
  </style>
</extension>
```

Enfin, il nous faut également penser à intégrer notre thème dans le point d'extension **applications** :

```
<extension target="org.nuxeo.theme.services.ThemeService" point="applications">
<application root="${org.nuxeo.ecm.contextPath}"
  template-engine="jsf-facelets">

  <negotiation>
    <strategy>nuxeo5</strategy>
    <default-engine>default</default-engine>
    <default-theme>esup/default</default-theme>
    <default-perspective>default</default-perspective>
  </negotiation>

  <!-- Cache control for theme resources (/nxthemes-lib/) -->
  <resource-caching>
    <lifetime>36000</lifetime>
  </resource-caching>

  <!-- Cache control for theme styles (/nxthemes-css/) -->
  <style-caching>
    <lifetime>900</lifetime>
  </style-caching>
</application>
</extension>

</component>
```

Application concrète

Le principe de fonctionnement ayant été rapidement abordé, voyons maintenant concrètement comment mettre en place notre nouveau thème.

Comme il y a des templates xhtml, des css et des images à intégrer en plus des points d'extension, il va nous falloir faire un plugin.

Pour ce faire, nous vous conseillons d'utiliser [Nuxeo IDE](#) avec Eclipse (choisir Nuxeo, Nuxeo Artifact, Nuxeo Plugin Project).

Notre plugin se présentera ainsi :

- src/main/resources/META-INF/MANIFEST.MF,
- src/main/java/packagePath => aucun code java ne sera ajouté pour personnaliser le thème
- pom.xml

Si nous reprenons les fichiers nécessaires, il nous faut un composant qui va regrouper tous nos points d'extension, le fichier esup-document.xml qui décrit la structure, les différents templates xhtml correspondant aux vues que nous avons créées. Il nous faut également stocker nos images (logo, favicon), nos palettes et nos css.

Au final, on retrouve un plugin avec l'arborescence suivante (quelques fichier exemples téléchargeables) :

- [src/main/resources/META-INF/MANIFEST.MF](#)
- [src/main/resources/OSGI-INF/deployment-fragment.xml](#)
- [src/main/resources/OSGI-INF/esup-theme-config.xml](#)*
- src/main/resources/themes/esup-document-management.xml
- src/main/resources/themes/css/esupStyle.css
- src/main/resources/themes/palette/default-backgrounds.properties
- src/main/resources/themes/palette/default-borders.properties
- src/main/resources/themes/palette/default-colors.properties
- src/main/resources/themes/palette/default-fonts.properties
- src/main/resources/web/nuxeo.war/incl/esup_header_template.xhtml
- [src/main/resources/web/nuxeo.war/incl/esup_header.xhtml](#)
- src/main/resources/web/nuxeo.war/incl/esup_footer_template.xhtml
- src/main/resources/web/nuxeo.war/incl/esup_footer.xhtml
- src/main/resources/web/nuxeo.war/incl/esup_logo.xhtml
- src/main/resources/web/nuxeo.war/img/esup_logo.png
- src/main/resources/web/nuxeo.war/icons/favicon.png

- `src/main/java/packagePath`
- `pom.xml`

Une fois le plugin achevé, il ne reste plus qu'à le déposer sous la forme d'un jar dans le répertoire plugins du template custom.

Redémarrez ensuite Nuxeo.



Attention, il existe de nombreuses contributions de thèmes dans Nuxeo : suivant les composants que vous installez (DAM, SC, user center, admin center), il vous faudra surcharger certainement plusieurs contributions pour que votre thème soit pris en compte partout.