

LUTECE : GUIDE INTÉGRATEUR

Published : 2015-09-15

License : GPL

INTRODUCTION

QU'EST CE QUE LUTECE ?



Lutece est un **logiciel libre** diffusé sous **licence BSD** par la Mairie de Paris depuis septembre 2002.

Lutece est écrit en **Java** et repose sur des briques logicielles Open Source faisant partie des standards (SpringFramework, Lucene, Freemarker, Ehcache, Apache-Commons, ...). Il a été accepté en 2014 au sein du consortium Open Source International OW2.

Lutece était à l'origine un CMS / Portail, mais a évolué au fil des années vers un Framework de développement d'applications Web. Cet outil est très largement utilisé par la Mairie de Paris pour la réalisation de ses sites, ses services numériques et ses applications métiers.

L'architecture très modulaire de **Lutece** permet une très grande souplesse dans l'intégration de fonctionnalités spécifiques à un portail. Les plugins, les modules, l'injection de dépendances sont autant de solutions pour réaliser des interfaces sur mesure tant sur le *Front Office* que sur le *Back Office*.

Lutece s'avère donc un outil complet prêt pour les plus grandes exigences et certaines normes. Si, au niveau de l'utilisation, il partage l'essentiel avec des CMS plus répandus, l'existence de plug-ins métier spécifiques en fait un outil hautement personnalisable en terme fonctionnel.

Lutece est ou a été utilisé par de prestigieuses références telles que MétéoFrance.fr, Notaires.fr, AtosWorldline.com, GeoPortail.gouv.fr, Paris.fr ou Marseille.fr.

La personnalisation graphique répond aux standards du Web HTML5 et CSS3 (utilisation à l'heure actuelle du framework Twitter Bootstrap), une interface graphique permet d'ajouter en ligne des styles de présentations variés ainsi que de nouveaux modèles de pages. Un moteur d'indexation puissant (Lucene) permet de faire des recherches sur tout le contenu du site, que ce soit sur les pages, les documents ou les fichiers des référentiels de contenu.

Ce livre a pour but d'aider l'intégrateur à mettre en place des instances de **Lutece**. Il part du principe que certaines connaissances sont déjà acquises comme des bases en ligne de commande, en HTML, CSS.

Pour compléter votre information au-delà de ce livre et rester en contact, voici une liste de liens à mettre dans vos favoris :

- Le site officiel : <http://fr.lutece.paris.fr>
- La démo en ligne (données réinitialisées toutes les 3 heures) : <http://dev.lutece.paris.fr/site-demo/>
- Le code source : <https://github.com/lutece-platform>
- Les versions eBook de ce présent livre et des manuels utilisateur et développeur : <http://www.flossmanualsfr.net>
- Les métriques Open Source du projet : <https://www.openhub.net/p/lutece>
- La documentation Maven du projet : <http://dev.lutece.paris.fr>
- Le Wiki contenant les Howto : <http://wiki.lutece.paris.fr>
- Le compte twitter pour les news générales : <https://twitter.com/LuteceNews>
- Le compte twitter pour les news de la fonderie : <https://twitter.com/LuteceFoundry>
- Le site Meetup : <http://www.meetup.com/Lutece/>
- L'image Docker du site de démo : <https://registry.hub.docker.com/u/lutece/lutece-demo/>

À PROPOS DE CE LIVRE

Ce livre a été rédigé durant un Booksprint de 5 jours en août 2015, organisé par Floss Manuals Francophone et l'équipe de **Lutece**. Les coauteurs de cet ouvrage sont : Magali Bouvat, Baptiste Gabourin, Cédric Gémy, Jon Harper, Laurent Hohl, Elysa Jouve, Magali Lemaire, Isabelle Lenain, Pierre Levy, Alexandre Mangot, Loïc Martinez, François-Eric Merlin, Yvan Michel, Jérôme Poirier, Marie-Pierre Roux, Élisabeth Saumard, Rémi Zara et Elisa de Castro Guerra.

Certains contenus proviennent de contributions plus anciennes réalisées par : Valérian Broussard, Géraud Dugé de Bernonville, Thibault Lassalle, Louis Lin, Vashista Nobaub, Vincent Vossier, Johann Goulay.



CHOISIR SES COMPOSANTS

- 1. LES INCONTOURNABLES**
- 2. LES FONCTIONS (ET PLUGINS) CLASSIQUES**

1. LES INCONTOURNABLES

Certaines fonctionnalités standard d'un site web, comme la recherche, la génération de formulaires ou encore les statistiques, sont fournies par des plugins dits "génériques", réutilisables sur tout type de site. Ce chapitre présente les plugins les plus communément utilisés sur les sites Lutece.

LES COMPOSANTS DE RECHERCHE ET DE RÉFÉRENCIEMENT

Les premiers plugins visent à ajouter un moteur de recherche interne et à améliorer le référencement du site auprès des moteurs de recherches externes :

- **Librairie Lucene (library-lucene)** : moteur de recherche qui permet l'indexation du contenu de l'application **Lutece**.
- **Plugin SEO (plugin-seo)** : permet de centraliser la gestion de l'ensemble des modules dans le but d'optimiser le référencement sur les moteurs de recherche avec les URL explicites, et de générer automatiquement le plan du site (avec plugin-sitemap) et le fichier robots.txt.
- **Plugin Searchstats (plugin-searchstats)** : permet d'obtenir des statistiques sur les recherches effectuées sur le site.
- **Outils Google (plugin-gtools)** : permet d'inclure dans les pages du site des éléments destinés aux services Google, tels que Google Analytics et Google Webmaster tools.

LES COMPOSANTS DE GESTION DE CONTENU

Pour la création et la gestion des contenus du site :

- **Plugin Document (plugin-document)** : indispensable dans le cas d'un site à contenu éditorial. Il permet la création et la publication de divers types de documents et listes de documents, gérés dans le back office.
- **Plugin Library (plugin-library)** : permet de configurer le mapping entre les attributs de Document et les feuilles de style XSL. Indispensable pour utiliser les fonctionnalités d'insertion de liens de ressources Lutece dans l'éditeur de texte riche (accessibles via le bouton ) , et indispensable si le plugin Document est installé.
- **Plugin HTML (plugin-html)** : permet d'ajouter des rubriques HTML éditables dans l'administration du site.
- **Plugin Pages filles (plugin-childpages)** : permet la création de portlets de navigation vers les pages filles de la page courante.

LES COMPOSANTS D'INTERACTION DE BASE

Pour créer des formulaires en ligne (formulaire d'inscription, formulaire de contact, ...), et proposer une protection antispam :

- **Plugin Formulaires (plugin-form)** : permet de créer des formulaires simples ou complexes, d'en recueillir les informations et de les exporter. Ce plugin dispose de plusieurs modules permettant notamment d'exporter les données de formulaires vers des annuaires (voir plus de détails dans le chapitre [Paramétrer un processus de gestion de demande](#)).
- **Plugin Contact (plugin-contact)** : Ce plugin permet de gérer plusieurs listes de contacts et d'afficher le formulaire d'envoi de l'e-mail.

LES COMPOSANTS TECHNIQUES

Enfin, pour afficher les versions des composants techniques du site :

- **Informations système (plugin-systeminfos)** Ce plugin permet d'afficher les informations système dans le back-office de l'application.

2. LES FONCTIONS (ET PLUGINS) CLASSIQUES

Ce chapitre présente quelques exemples de plugins couvrant les besoins classiques d'un site web.

AUTHENTIFICATION FRONT OFFICE

Cette fonctionnalité permet de gérer l'authentification sur le front office d'un site, dans le cadre d'un site partiellement ou totalement identifié (portail de service, extranet, intranet, outil métier, ...). Le plugin principal MyLutece permet d'associer autant de plug-in d'authentification que nécessaire. Il est possible de cumuler les types d'authentification sur un site, d'utiliser des types d'authentification standards, ou encore de créer son propre module d'authentification correspondant à ses besoins particuliers.

- **Plugin MyLutece (plugin-mylutece)** : permet de brancher les modules d'authentification;
- **Module MyLutece LDAP (module-mylutece-ldap)** : authentification via un annuaire ldap;
- **Module MyLutece CAS (module-mylutece-cas)** : authentification via le système d'authentification unique (SSO) CAS (Central Access Service);
- **Module MyLutece FranceConnect (module-mylutece-franceconnect)** : authentification via le système d'authentification FranceConnect promu par les services informatiques de l'Etat;
- **Module MyLutece OAuth (module-mylutece-oauth)** : authentification via le protocole OAuth utilisé par des services tels que Google ou autre;
- ...

EXTENSIONS (COMMENTAIRES, VOTRE AVIS, VUES, ...)

Ce type de plugins permet d'intégrer des fonctionnalités complémentaires aux types de ressources d'un plugin. Il s'agit de pouvoir effectuer des actions en front-office, intégrables de manière modulaire.

Le plugin principal de cette fonctionnalité permet de faire fonctionner tous les modules associés. Le paramétrage de ces outils est décrit dans le chapitre [Paramétrer les extensions](#)

- **Plugin Extend (plugin-extend)** : permet d'insérer des extensions de contenu ;
- **Module Commentaires (module-extend-comment)** : permet de commenter le contenu ;
- **Module Votre Avis (module-extend-feedback)** : permet de donner un avis par mail sur le contenu publier ;
- **Module Réseaux sociaux (module-extend-opengraph)** : permet d'insérer les boutons de partage sur les réseaux sociaux.

FAQ

- **Plugin Helpdesk (plugin-helpdesk)** : permet de créer une FAQ classique avec questions-réponses paramétrables dans le back office.

GESTION D'AVATAR

La gestion d'avatar est basée sur le service internet Gravatar.

Un plugin Lutece permet de gérer son propre serveur d'avatar, utilisable par exemple dans les plugins participatifs, pour associer un avatar aux commentaires des internautes.

- **Plugin avatar (plugin-avatar)** : serveur gravatar;
- **Plugin avatarserver (plugin-avatarserver)** : gestion interne du serveur d'avatar.

GESTION DES LIBELLÉS SUR LE SITE

Cette fonctionnalité permet d'insérer des libellés n'importe où sur un site en front-office, la saisie et la mise à jour des libellés étant centralisée dans une interface commune en back office.

L'intégration de cette fonctionnalité est décrite dans le chapitre [Gérer les libellés](#).

- **Plugin Sitelabels (plugin-sitelabels)** : ajoute l'interface d'administration des libellés.

RECHERCHE

- **Plugin Solr (plugin-solr)** : s'interface avec Lucene et permet des fonctionnalités de recherche avancées, l'indexation de contenu avancée, l'autocomplétion, la recherche géolocalisée sur une carte.
- **Plugin Solr Server (plugin-solrserver)** : installe un serveur Solr embarqué dans la webapp, permettant d'utiliser le plugin Solr.

SÉCURITÉ

- **Plugin Captcha (plugin-captcha)** : intégration de captcha sur les formulaires.

STATISTIQUES

- **Plugin Piwik (plugin-piwik)** : permet de bénéficier de statistiques avancées sur la fréquentation du site, moyennant l'installation d'un serveur [Piwik](#) à part.
- **Plugin Dataviz (plugin-dataviz)** : permet d'obtenir des statistiques sur les données du site.

THÈMES

- **Plugin Thème (plugin-theme)** : gestion des thèmes graphiques de l'application.

ASSEMBLER UN SITE

- 3. PRÉREQUIS TECHNIQUES**
- 4. COMMENT FONCTIONNE L'ASSEMBLAGE ?**
- 5. CHOISIR DES COMPOSANTS SELON LE TYPE DE SITE**
- 6. INSTALLER LUTECE**

3. PRÉREQUIS TECHNIQUES

Lutece est un projet écrit en Java J2EE, il nécessite l'utilisation d'outils du monde Java.

Pour **assembler** un site **Lutece** il faut installer les logiciels suivants :

- **Apache Maven** v3 ou supérieur : Maven permet de construire un site à partir de ses sources. L'utilisation de Maven permet de générer les sites complets - en choisissant les différents composants Lutece- à partir de la plate-forme de développement Lutece (visitez le site officiel de [Apache Maven](#)).
- **Apache Ant** : outil d'automatisation de tâche, qui va permettre de construire et d'initialiser la base de données d'un site à partir des scripts d'initialisation inclus dans le site construit par Maven (site officiel de [Apache Ant](#)).

Pour **déployer** et **héberger** un site **Lutece** il faut les briques logicielles suivantes :

- Une **machine virtuelle Java** : par exemple JDK 1.7 ou supérieur.
- Un **serveur de type container de servlet** : [Tomcat 7](#) ou supérieur, [Glassfish](#), [Jetty](#).
- Un **serveur de base de données relationnelle** : [MySql](#), [MariaDB](#), [PostgreSQL](#).



Apache Tomcat



4. COMMENT FONCTIONNE L'ASSEMBLAGE ?

Assembler un site Lutece, en quoi cela consiste ?

Lutece utilise [Maven](#) pour décrire un site à partir d'un simple fichier, le fichier **pom.xml**. Dans ce fichier on donne la liste des plug-ins et des composants souhaités. À partir de là, il est facile de générer le site complet prêt à être déployé sur un serveur JEE.

Assembler, c'est donc traduire les fonctionnalités souhaitées en autant de composants techniques (plug-ins) **Lutece** et d'écrire le **pom.xml** pour pouvoir préparer le site qui sera alors techniquement prêt à être déployé.

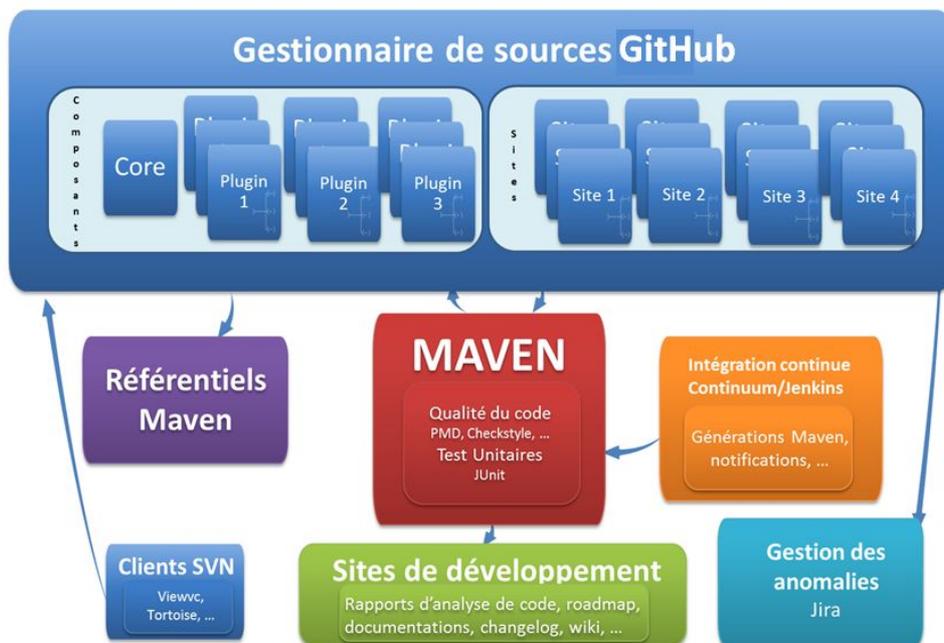
Avec cette méthode, nous allons aussi pouvoir gérer les éléments graphiques (css, images, js, modèles de pages, feuilles XSL de présentation des rubriques) et les configurations sur un outil de gestion de version, GIT par exemple. La procédure d'assemblage va alors générer non seulement un site techniquement prêt à fonctionner, mais surtout un site complètement paramétré et intégré graphiquement sur la plateforme d'hébergement choisie.

Assembler permet donc d'automatiser, de limiter les risques d'erreurs, de profiter facilement des dernières versions depuis la plateforme technique **Lutece** et des dernières évolutions graphiques pour avoir un site toujours à jour.

LA PLATE-FORME LUTECE

Le développement de Lutece est organisé autour d'une plate-forme d'intégration continue. Les sources du projet **Lutece** sont accessibles en ligne sur [GitHub](#).

L'accès à cette plate-forme se fait à travers un portail de développement communautaire (<http://dev.lutece.paris.fr>). Il permet de contrôler les processus d'intégration continue, d'accéder à l'outil de suivi des demandes ([JIRA](#)), au référentiel de source public [GitHub](#), au [Wiki](#), aux rapports de qualité du code, d'automatiser les procédures de développement et de distribution des sites (Jenkins), d'encadrer la gestion des composants **Lutece** (plug-ins) en projets indépendants, de contrôler le respect des normes de développement et la qualité du code.



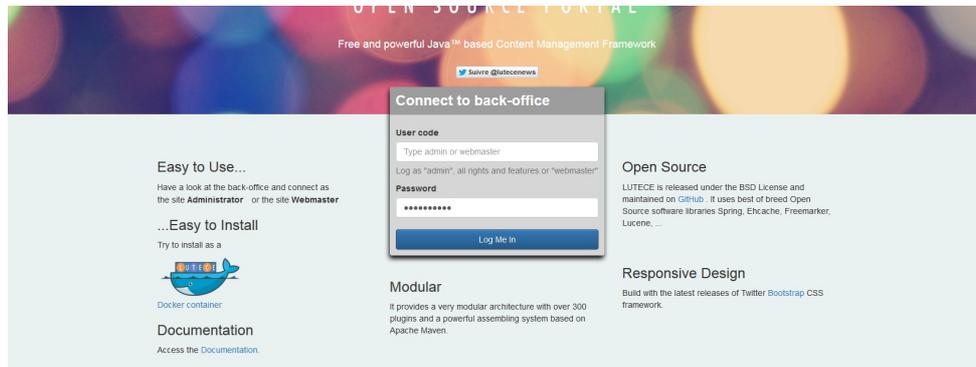
Le code source applicatif de Lutece est mis à jour dans le référentiel (« pushé ») par les développeurs au fur et à mesure des développements sur l'outil de gestion des sources ([GitHub](#)). La construction des packages binaire ("build") est pilotée par l'outil d'intégration Jenkins, à l'aide de "goals" [Maven](#) spécifiquement développés pour **Lutece** :

- **lutece:exploded** : pour compiler un composant pendant la phase de développement (voir Guide de développement Lutece);
- **lutece:assembly** : pour créer un package binaire d'un composant, pour ajouter une fonctionnalité à un site existant;
- **lutece:site-assembly** : la commande qui nous intéresse ici pour créer un site à partir d'un fichier pom.xml de site.

Jenkins se charge entre autres de construire toutes les nuits les packages de l'ensemble des projets **core** et **plug-ins Lutece**.

L'ensemble des composants est mis à jour continuellement au fur et à mesure de l'avancée des développements, il est possible de visualiser en ligne les dernières fonctionnalités réalisées sur le site de démo (<http://dev.lutece.paris.fr/site-demo>).





GÉNÉRER UN SITE EN ASSEMBLANT LES COMPOSANTS

Chaque projet ou sous-projet est configuré par un fichier nommé pom.xml (POM : Project Object Model), situé à la racine du projet.

Celui-ci contient les informations nécessaires à Maven pour traiter le projet (nom du site, numéro de version, dépendances vers les plug-ins composants le site, etc.).

Le pom.xml d'un site **Lutece** liste donc l'ensemble des composants, chacun couvrant des fonctionnalités particulières, nécessaires pour réaliser un site du type voulu : éditorial, portail de service ou encore participatif.

Le fichier POM d'un site contient au minimum les informations suivantes :

- La déclaration du POM parent pour tous les sites (**lutece-site-pom**)
- Les déclarations des repository Maven (« Snapshot » facultatif)
- La déclaration des plug-ins qui composent le site sous forme de dépendances

Exemple de fichier pom.xml pour un site de formulaire paramétrable simple :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
v4_0_0.xsd">
  <parent>
    <artifactId>lutece-site-pom</artifactId>
    <groupId>fr.paris.lutece.tools</groupId>
    <version>2.0.4</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>
  <groupId>fr.paris.lutece</groupId>
  <artifactId>site-mysite</artifactId>
  <packaging>lutece-site</packaging>
  <version>1.0.0-SNAPSHOT</version>
  <name>My site</name>

  ....

  <dependencies>
    <dependency>
      <groupId>fr.paris.lutece</groupId>
      <artifactId>lutece-core</artifactId>
      <version>5.1.1</version>
      <type>lutece-core</type>
    </dependency>
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>plugin-form</artifactId>
      <version>4.1.1</version>
    </dependency>
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>module-form-exportdatabase</artifactId>
      <version>1.0.3</version>
    </dependency>
  </dependencies>
</project>
```

Dans les exemples donnés, les numéros de version sont donnés à titre indicatif. Il s'avère important de vérifier les versions disponibles pour les plug-ins, mais aussi pour leurs dépendances et de modifier en fonction.

Les plug-ins Lutece sont des composants qui permettent d'ajouter à Lutece des fonctionnalités. Chaque plug-in est un ensemble de fonctionnalités, par exemple, les formulaires en ligne seront intégrés par le plugin form, les lettres d'informations par le plugin newsletter...

Les modules sont des plug-ins de plug-ins. Ils permettent d'ajouter des fonctionnalités spécifiques à un plug-in existant. On peut citer en exemple la fonctionnalité d'export des données d'un formulaire dans une base de données tierce avec le module « module-form-exportdatabase ».

Pour plus d'informations sur le fichier pom, reportez-vous à la [documentation de Maven](#).

Pour ajouter des fonctionnalités à notre site de formulaires paramétrables simple, il faut ajouter les plug-ins et modules correspondants aux améliorations souhaitées :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
v4_0_0.xsd">
  <parent>
    <artifactId>lutece-site-pom</artifactId>
    <groupId>fr.paris.lutece.tools</groupId>
    <version>2.0.4</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>
  <groupId>fr.paris.lutece</groupId>
  <artifactId>site-mysite</artifactId>
  <packaging>lutece-site</packaging>
  <version>1.0.0-SNAPSHOT</version>
  <name>My site</name>

  ....

  <dependencies>
    <dependency>
      <groupId>fr.paris.lutece</groupId>
      <artifactId>lutece-core</artifactId>
      <version>5.1.1</version>
      <type>lutece-core</type>
    </dependency>
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>plugin-form</artifactId>
      <version>4.1.1</version>
    </dependency>
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>module-form-exportdatabase</artifactId>
      <version>1.0.3</version>
    </dependency>
    <!-- Ajout du plugin de gestion du captcha -->
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>plugin-captcha</artifactId>
      <version>2.1.0</version>
    </dependency>
    <!-- Ajout du module du type de captcha voulu -->
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>module-captcha-jcaptcha</artifactId>
      <version>2.1.11</version>
    </dependency>
    <!-- Ajout du plugin de visualisation des statistiques de fréquentation du site -->
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>plugin-piwik</artifactId>
      <version>1.0.1</version>
    </dependency>
    <!-- Ajout du plugin de contrôle des champs par expression régulière -->
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>plugin-regularexpression</artifactId>
      <version>3.0.2</version>
    </dependency>
    <!-- Ajout du plugin de réécriture d'url -->
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>plugin-seo</artifactId>
      <version>3.2.0</version>
    </dependency>
    <!-- Ajout du plugin de visualisation des informations du serveur de déploiement -->
    <dependency>
      <groupId>fr.paris.lutece.plugins</groupId>
      <artifactId>plugin-systeminfo</artifactId>
      <version>3.1.0</version>
    </dependency>
  </dependencies>
</project>
```

Une fois l'assemblage du POM terminé, celui-ci peut être compilé à l'aide des commandes Maven spécialement créées pour **Lutece**.

La commande de génération d'un site à partir de ce type de POM est la suivante :

```
mvn -U lutece:clean lutece:site-assembly
```

La compilation va permettre d'aller chercher les composants sur le repository, de les assembler dans un répertoire *target* qui va contenir l'ensemble des répertoires et fichiers nécessaires au bon fonctionnement du site. Le répertoire du site peut ensuite être déposé dans le répertoire webapp de Tomcat par exemple pour déploiement (voir [Installer Lutece](#)).

IMPORTANT Pour un site en recette ou en production, ne pas utiliser des plages de versions pour les dépendances. Les versions explicites doivent être utilisées. Il est conseillé de supprimer la déclaration du référentiel « Snapshot » et d'utiliser un cache maven local séparé ne contenant aucun « Snapshot » pour s'assurer du contrôle parfait de toutes les versions des composants.

La version complète du fichier POM est disponible à titre d'exemple en [Annexe 1](#)

ATTENTION les versions des composants sont à mettre à jour, le core et les plug-ins Lutece évoluent en continu.

Les outils ci-dessous permettent de réaliser un site ou POM à jour de manière simplifiée.

GESTION DES DÉPENDANCES

Plusieurs outils sont disponibles pour l'élaboration de POM de sites :

- [AppStore](#) : Fournit des POM de sites standards intégrant les dernières versions de chacun des plug-ins.
- [Liste des composants](#) : Fournit la liste de tous les composants (plug-ins, modules, bibliothèques) en indiquant les dernières versions Release et Snapshot ainsi qu'un accès rapide aux évolutions et anomalies en cours.
- [Upgrade des dépendances](#) : Fournit la liste des dépendances dans leur dernière version au format XML du POM pour une liste de plug-ins donnée.

[Le dernier outil](#) permet de générer un fichier POM avec les composants à jour de manière simple et rapide, simplement en listant les plug-ins nécessaires dans son site.

Petit exemple d'utilisation pour réaliser le POM ci-dessus :

1. insérer la liste des composants nécessaire pour son site
2. sélectionner le type de sortie souhaitée et cliquer sur Générer
3. copier/coller le résultat produit dans le bloc de droite, et le coller dans un nouveau fichier, l'enregistrer sous le nom pom.xml

LUTECE Source code Documentation

Mise à niveau des dépendances

Dépendances en entrée
Coller dans le champ ci-dessous la liste des plugins pour lesquels vous souhaitez construire les déclarations de dépendances incluant les dernières versions "releasées".

- plugin-form
- module-form-exportdatabase
- plugin-captcha
- module-captcha-jcaptcha
- plugin-piwik
- plugin-regularexpression
- plugin-seo
- plugin-systeminfo

1 Ajouter la liste des composants en indiquant leur nom

Format de sortie : Texte XML Nouveau POM **Générer**

Dépendances en sortie

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/maven-v4_0_0.xsd">
  <parent>
    <artifactId>lutece-site-pom</artifactId>
    <groupId>fr.paris.lutece.tools</groupId>
    <version>2.0.4</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>
  <groupId>fr.paris.lutece</groupId>
  <artifactId>site-mysite</artifactId>
  <packaging>lutece-site</packaging>
  <version>1.0.0-SNAPSHOT</version>
```

3 Copier-coller le fichier POM généré

Cet outil vous permettra d'obtenir un fichier pom.xml avec les dernières versions à jour des composants.

5. CHOISIR DES COMPOSANTS SELON LE TYPE DE SITE

Concevoir un site avec Lutece est une histoire de choix, à la fois de fonctionnalités et de versions. Dans ce chapitre nous allons mettre en place des exemples de différents types de sites en proposant pour chacun la liste des plug-ins et modules à choisir pour générer un site immédiatement opérationnel et qui répondent à l'objectif que l'on s'est fixé à savoir : *informer, proposer et participer*.

INFORMER

Un site d'information nécessitera des composants permettant de mettre en place les fonctionnalités de publication éditoriale.

Un certain nombre de ces fonctionnalités sont incluses dans le core de Lutece, qui est à l'origine un outil de CMS (Content Management System) avant de devenir un framework de développements web.

Ainsi les fonctionnalités de gestion des utilisateurs, d'administration des pages du site, de positionnement de rubrique de contenu dans les pages par exemple sont présentes dans un site qui ne contiendrait que le composant core, c'est-à-dire le composant minimum indispensable pour générer un site Lutece.

Il s'agit ensuite d'enrichir son site avec toutes les fonctionnalités qui seraient nécessaires pour répondre à son besoin, en incluant la liste des composants nécessaires dans son POM de site.

Par exemple, un site éditorial permettant de publier de l'information variée serait généré par un pom.xml ayant entre autres les composants suivants :

Pour la navigation sur le site, des plug-ins permettent de gérer les pages filles, d'insérer des rubriques personnalisables, des menus, d'insérer via l'éditeur de texte riche :

```
<!-- Plugin de gestion des rubriques de pages filles du site -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-childpages</artifactId>
  <version>4.0.1</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Plugin de gestion de menus de navigation du site front office -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-menus</artifactId>
  <version>2.0.4</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Plugin d'insertion de liens vers des pages du site -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-linkpages</artifactId>
  <version>4.0.3</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Plugin d'insertion dans l'éditeur de texte de liens vers les pages du site -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-pagelinkservice</artifactId>
  <version>3.0.3</version>
  <type>lutece-plugin</type>
</dependency>
```

Pour la saisie de contenus simples et directs dans la page en BO, un éditeur HTML riche, avec un plug-in intégré d'insertion de contenu provenant de différents contenus, provenant de Lutece ou d'interface extérieure (document, pages filles, intégration d'une carte, d'un slider, d'une vidéo...):

```
<!-- Plugin d'insertion de contenu directement dans une rubrique du site via un éditeur html -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-html</artifactId>
  <version>3.1.0</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Plugin d'insertion de contenu dans une page du site via un éditeur html -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-htmlpage</artifactId>
  <version>4.0.1</version>
  <type>lutece-plugin</type>
</dependency>
```

Pour la saisie de contenus structurés, le plug-in document et les plug-ins/modules associés :

```
<!-- Plugin de création de contenu éditorial structuré -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-document</artifactId>
  <version>4.0.1</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Plugin d'insertion en masse de fichiers dans un espace de document (par exemple toutes
```

```

les photos d'un répertoire de son disque dure pour constituer un album en ligne) -->
  <dependency>
    <groupId>fr.paris.lutece.plugins</groupId>
    <artifactId>plugin-files2docs</artifactId>
    <version>1.0.6</version>
    <type>lutece-plugin</type>
  </dependency>
<!-- Plugin de gestion de la newsletter (contenu type document et saisie html directe -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-newsletter</artifactId>
  <version>3.0.2</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Plugin d'insertion des documents dans la newsletter -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>module-newsletter-document</artifactId>
  <version>1.0.2</version>
  <type>lutece-plugin</type>
</dependency>

```

Pour la gestion d'un agenda, avec possibilité d'associer du contenu documentaire aux événements

```

<!-- Plugin de gestion d'un agenda -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-calendar</artifactId>
  <version>4.1.0</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Plugin d'association de document éditoriaux à des événement de l'agenda -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>module-documenttocalendar</artifactId>
  <version>1.1.1</version>
  <type>lutece-plugin</type>
</dependency>

```

Il est possible d'ajouter des « extensions de contenu », qui permettent d'ajouter des commentaires, votes, demander l'avis sur une ressource, des barres de réseaux sociaux...

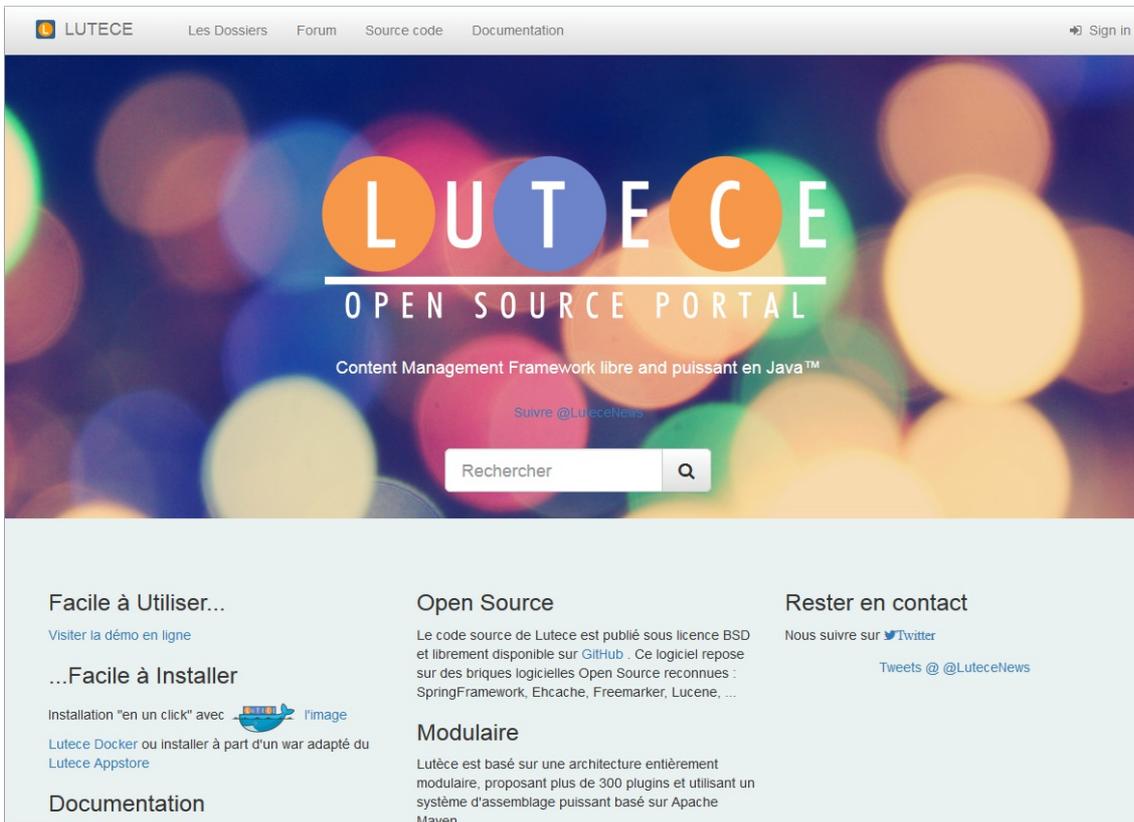
```

<!-- Plugin des extensions sur le FO -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-extend</artifactId>
  <version>1.2.1</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Plugin d'insertion d'un extension de type commentaire sur le FO -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>module-extend-comment</artifactId>
  <version>1.1.4</version>
  <type>lutece-plugin</type>
</dependency>

```

Le fichier complet est accessible en [Annexe 2](#) (à renommer en fichier pom.xml)

Voici un exemple de site purement éditorial



PROPOSER

Ce type de site permet de proposer des services pour un usager du site.

Typiquement il permet de proposer des télé-services dématérialisés, de manière paramétrable, avec suivi de ses demandes par l'utilisateur et traitement des demandes dans le back-office.

Dans le cadre d'une Administration, ce type de site est couramment utilisé pour proposer des sites de type « Appels à projets », dématérialisation de procédures administratives, accessibles à partir d'un guichet en ligne (de type Portail Citoyen par exemple)

Les plug-ins de proposition de services dématérialisés sont soit « clé en main », à paramétrer, soit des développements spécifiques rendant un service précis, en général en interaction avec des applications métiers d'un Système d'Information.

Le Guichet d'accès est constitué d'un ensemble de plug-ins permettant à un utilisateur connecté d'accéder à la liste des services en ligne, à la liste de ses demandes en cours et leur état d'avancement, éventuellement à ses demandes en état brouillon. Plusieurs plug-ins techniques permettent à ce composant d'interagir avec d'autres plug-ins, ou des applications métiers, par le biais de web service Rest, pour les remontées de notifications ou des statuts d'avancement de la demande par exemple.

```

<!-- Plugin interface guichet -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-crm</artifactId>
  <version>1.0.10-SNAPSHOT</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Module technique de communication Rest avec d'autres services -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>module-crm-rest</artifactId>
  <version>1.0.5-SNAPSHOT</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Module technique de communication Rest avec le plugin Form, pour accéder au formulaire en
mode brouillon notamment -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>module-crm-form</artifactId>
  <version>2.0.1-SNAPSHOT</version>
  <type>lutece-plugin</type>
</dependency>
<!-- Module technique de communication Rest avec le plugin Workflow, pour notifier
l'utilisateur selon l'état d'avancement de sa demande -->
<dependency>

```

```

    <groupId>fr.paris.lutece.plugins</groupId>
    <artifactId>module-workflow-notifycrm</artifactId>
    <version>3.0.1-SNAPSHOT</version>
    <type>lutece-plugin</type>
  </dependency>

```

Les téléseices peuvent être mis en ligne avec différents types de composants. Le plus facile à utiliser est celui basé sur le plugin-form, qui permet de paramétrer dynamiquement un formulaire simple (voir documentation utilisateur, chapitre gérer les formulaires). Mais des téléseices rendant des services plus spécifiques, packagés sous forme de plug-in dédié peuvent être ajoutés au sein d'un même site. Cela peut permettre par exemple de composer un formulaire comprenant des listes déroulantes valorisées par une application métier par exemple. Les principes de développement de ces plug-ins spécifiques sont accessibles dans le manuel de développement Lutece, chapitre Développement de composants Lutece.

```

<!-- Plugin de formulaires paramétrables -->
</dependency>
  <dependency>
    <groupId>fr.paris.lutece.plugins</groupId>
    <artifactId>plugin-form</artifactId>
    <version>4.1.1-SNAPSHOT</version>
    <type>lutece-plugin</type>
  </dependency>
<!-- Plugin spécifique de type xpage -->
</dependency>
  <dependency>
    <groupId>fr.paris.lutece.plugins</groupId>
    <artifactId>plugin-stationnement</artifactId>
    <version>[1.0.0-SNAPSHOT,)</version>
    <type>lutece-plugin</type>
  </dependency>
<!-- Plugin spécifique de type xpage -->
</dependency>
  <dependency>
    <groupId>fr.paris.lutece.plugins</groupId>
    <artifactId>plugin-quotientfamille</artifactId>
    <version>[1.0.0-SNAPSHOT,)</version>
    <type>lutece-plugin</type>
  </dependency>

```

Les données des téléseices peuvent être stockées/traitées par des plug-ins génériques avec lesquels ils interagissent via des modules de communication entre les plug-ins

```

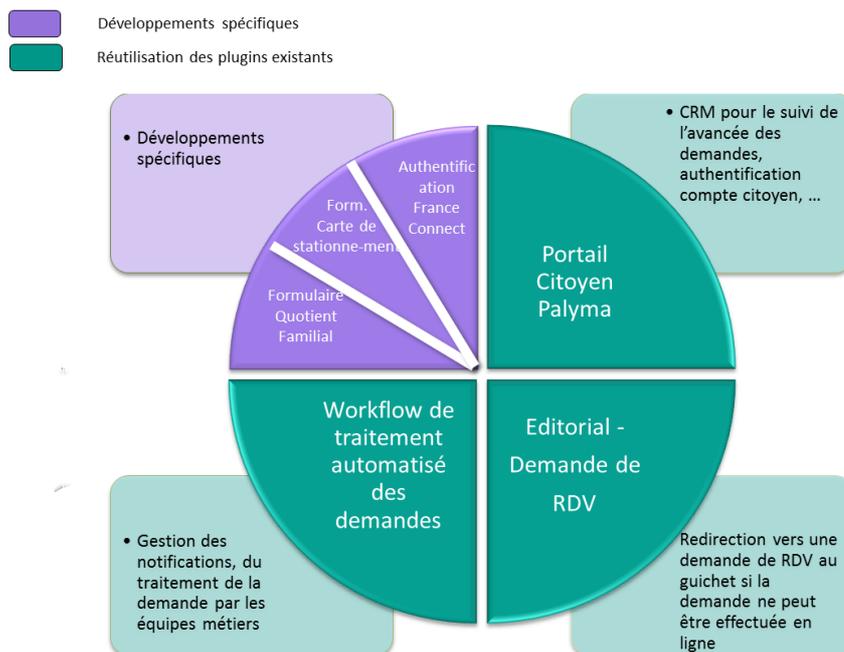
<!-- Plugin de gestion d'une liste d'états - transition , utilisable dans différents types de
plugins Lutece -->
</dependency>
  <dependency>
    <groupId>fr.paris.lutece.plugins</groupId>
    <artifactId>plugin-workflow</artifactId>
    <version>4.2.1-SNAPSHOT</version>
    <type>lutece-plugin</type>
  </dependency>
<!-- Plugin de gestion de données Lutèce -->
</dependency>
  <dependency>
    <groupId>fr.paris.lutece.plugins</groupId>
    <artifactId>plugin-directory</artifactId>
    <version>4.0.1-SNAPSHOT</version>
    <type>lutece-plugin</type>
  </dependency>
<!-- Module d'insertion des données du plugin form dans le plugin directory -->
</dependency>
  <dependency>
    <groupId>fr.paris.lutece.plugins</groupId>
    <artifactId>module-form-exportdirectory</artifactId>
    <version>3.0.1-SNAPSHOT</version>
    <type>lutece-plugin</type>
  </dependency>
<!-- Module de génération d'un pdf d'une fiche directory-->
</dependency>
  <dependency>
    <groupId>fr.paris.lutece.plugins</groupId>
    <artifactId>module-directory-pdfproducer</artifactId>
    <version>1.1.2-SNAPSHOT</version>
    <type>lutece-plugin</type>
  </dependency>

```

Le fichier complet d'exemple d'un site de type guichet avec suivi des demandes est accessible en [Annexe 3](#) (à renommer en fichier pom.xml). Il s'agit du site réalisé dans le cadre d'un Hackathon organisé par la DISIC sur le module d'authentification France Connect. Il est composé des plug-ins permettant de réaliser un guichet dématérialisé, de téléseices métiers, de composants éditoriaux permettant d'enrichir le site...

Exemple d'architecture d'un site de téléseices

Un exemple de Guichet de téléservices avec Lutèce



Les interfaces front office de ce type de site permettent à un usager par exemple de conserver un brouillon de sa demande tant qu'elle n'a pas été soumise à l'administration :

MAIRIE DE PARIS
GUICHET DES PROFESSIONNELS DE PARIS
 Yves Michel | Déconnexion
 Dernière connexion : 13/11/2013 13:57:37

accueil > Gestion des relations citoyens
 Accueil | Mes demandes | **Mes brouillons** | Mon compte

Mes brouillons

	Date de modification	Statut	Actions	Notification
Demande de terrasse ou étalage	09/10/13 09:53	Nouveau	[Icones]	
Demande de terrasse ou étalage	08/10/13 18:08	Nouveau	[Icones]	
Demande de terrasse ou étalage	27/09/13 10:35	En cours de saisie	[Icones]	
Demande d'enseigne	12/09/13 11:33	Nouveau	[Icones]	
Demande de terrasse ou étalage	12/09/13 11:30	Nouveau	[Icones]	
Demande d'enseigne	05/09/13 14:13	Nouveau	[Icones]	
Demande de terrasse ou étalage	05/09/13 14:13	En cours de saisie	[Icones]	
Demande de terrasse ou étalage	21/08/13 14:15	Nouveau	[Icones]	
Demande de terrasse ou étalage	21/08/13 13:33	Nouveau	[Icones]	
Demande de terrasse ou étalage	12/08/13 16:37	Nouveau	[Icones]	

Recherche
 Type de demande
 Date de modification
 Notification
 Faire une recherche sur le contenu des notifications.
 Rechercher

PARIS.fr MAIRIE DE PARIS
 Réseaux sociaux (f, t)
 Infos (Contactez nous, Mentions légales, Foire aux questions)

Ou encore d'accéder de manière centralisée à la liste de ses demandes à l'administration, de télécharger des récapitulatifs pdf, d'accéder à l'ensemble de ses notifications, de compléter ses demandes à l'initiative du gestionnaire back-office, ...

MAIRIE DE PARIS

GUICHET DES PROFESSIONNELS DE PARIS

Yves Michel
Déconnexion

Dernière connexion: 13/11/2013 13:57:37

accueil > Gestion des relations citoyens

Accueil
Mes demandes
Mes brouillons
Mon compte

Mes demandes

Précédent 1 2 Suivant

Nombre par page: 10 - Nombre d'éléments: 11

	Date de modification ^ v	Statut	Notification ^ v
Demande d'enseigne	09/10/13 11:07	Demande en cours d'instruction	<input type="checkbox"/>
Demande de cendrier mobile	05/09/13 14:14	Nouvelle demande	<input type="checkbox"/>
Demande de renseignement sur sous sol	20/08/13 16:15	En attente de paiement	<input type="checkbox"/>
Demande d'enseigne	13/06/13 12:04	Nouvelle demande	<input type="checkbox"/>
Demande d'enseigne	13/06/13 11:23	Nouvelle demande	<input type="checkbox"/>
Demande de renseignement sur sous sol	12/06/13 15:24	En cours	<input type="checkbox"/>
Demande d'enseigne	31/05/13 10:28	Nouvelle demande	<input type="checkbox"/>
Demande d'enseigne	14/05/13 09:47	Nouvelle demande	<input type="checkbox"/>
Demande d'enseigne	25/04/13 17:20	Nouvelle demande	<input type="checkbox"/>
Demande d'enseigne	16/04/13 11:23	Nouvelle demande	<input type="checkbox"/>

Précédent 1 2 Suivant

Nombre d'éléments: 11

Recherche

Type de demande

Date de modification

Notification

Faire une recherche sur le contenu des notifications.

Rechercher

Réseaux sociaux

Infos

[Contactez nous](#)
[Mentions légales](#)
[Foire aux questions](#)

PARTICIPER

Ce type de site permet de proposer des services de participation des usagers, citoyens, acteurs d'un projet. Typiquement il permet de proposer des sites où les internautes pourront donner leur avis sur un sujet, réagir à ces sujets et voter.

Le plug-in concerné peut être présenté sous la forme de forum ou encore d'outil participatif pour faire contribuer les internautes en leur permettant de faire des propositions sur un sujet précis, avec possibilité de commenter et voter sur chaque proposition.

```

<!-- Plugin permettant de réaliser un outil de proposition/votes/commentaires par les
internautes, ou encore un forum -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-digglike</artifactId>
  <version>2.3.1</version>
</dependency>

```

Il est possible d'ajouter des « extensions de contenu » permettant d'associer à tout type de contenu des possibilités d'interaction telles que les commentaires, votes, etc., sur des plug-ins standards de Lutece ou sur des plug-ins spécifiques (voir le Guide de développement Lutece - Chapitre Extensions de contenus, et le chapitre [paramétrer les extensions](#) de ce présent guide)

```

<!-- Plugin et modules permettant d'ajouter des interactions des internautes sur du contenu de
tout type -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-extend</artifactId>
  <version>1.3.0</version>
</dependency>
<!-- Module de commentaires -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>module-extend-comment</artifactId>
  <version>1.1.4</version>
</dependency>
<!-- Module d'avis-->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>module-extend-feedback</artifactId>
  <version>1.0.3</version>
</dependency>
<!-- Module d'insertion d'une barre de réseaux sociaux -->

```

19

```

<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>module-extend-opengraph</artifactId>
  <version>1.2.0</version>
</dependency>

```

Le plug-in Wiki permet d'intégrer un wiki à son site, enrichissant ainsi le contenu d'un espace de contributions partagées

```

<!-- Plugin Wiki -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-wiki</artifactId>
  <version>2.1.0</version>
</dependency>

```

Ces modules sont en général utilisés dans le cadre de site en partie au moins authentifiés

```

<!-- Plugin et module d'authentification des internautes -->
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>plugin-mylutece</artifactId>
  <version>3.3.0</version>
</dependency>
<dependency>
  <groupId>fr.paris.lutece.plugins</groupId>
  <artifactId>module-mylutece-database</artifactId>
  <version>4.0.0</version>
</dependency>

```

Le fichier complet est accessible en [Annexe 4](#) (à renommer en fichier pom.xml)

Exemple de Wiki

The screenshot shows a Wiki page with the following content:

Titre de la page : Génération de code avec PluginWizard

Contenu de la page :

= Génération de code avec PluginWizard =

PluginWizard est un plugin Lutece permettant de générer les principaux fichiers d'un plugin Lutece.

- * Les classes Java de la couche métier
- * Les classes Java de la couche présentation
- * Les classes Java des XPage
- * Les JSP de l'interface d'administration
- * Les templates HTML
- * Les fichiers ressources d'internationalisation (i18n)
- * Le fichier properties de configuration du plugin
- * Le fichier de contexte de Spring
- * Le fichier XML de définition du plugin
- * Les scripts SQL de création et d'initialisation de la base de données
- * Le fichier pom.xml pour construire le projet avec Maven
- * Les fichiers de documentation au format xDoc

[[http://dev.lutece.paris.fr/site-demo/jsp/site/Portal.jsp?page=pluginwizard | Accès à PluginWizard sur le site de démo]]

Commentaire concernant les modifications :

Droits d'accès

Rôle pour voir la page : No restriction

Rôle pour modifier la page :

6. INSTALLER LUTECE

Une fois l'assemblage effectué (voir chapitre précédent) il faut :

- Déployer le .war qui se trouve dans le répertoire *target/* de l'application
- Paramétrer la connexion à la base de données
- Initialiser la base de données

DÉPLOYER LE WAR

Déposer le fichier .war contenant l'application sur votre serveur d'applications (local ou distant), par exemple dans le répertoire nommé *webapps* d'un serveur *Tomcat*.

PARAMÉTRER LA CONNEXION À LA BASE DE DONNÉES

Les informations de connexion à la base de données se trouvent dans le fichier *WEB-INF/conf/db.properties*. Les valeurs par défaut doivent être modifiées, si cela n'a pas été fait en amont de l'assemblage du site.

Le fichier *db.properties* se présente comme suit :

```
#####
# Configuration file for Lutece to parameterize connections pools to databases
#####
# Drivers

#portal.poolservice is not a mandatory property
portal.poolservice=fr.paris.lutece.util.pool.service.LuteceConnectionService
portal.driver=org.gjt.mm.mysql.Driver
portal.url=jdbc:mysql://localhost/lutece?autoReconnect=true&useUnicode=yes&characterEncoding=utf8
portal.user=
portal.password=
portal.initconns=2
portal.maxconns=50
portal.logintimeout=2
portal.checkvalidconnections=SELECT 1
# <pool>.dialect is an optional property to specify the dialect for JPA provider.
#portal.dialect=org.hibernate.dialect.MySQLDialect
```

Les valeurs à modifier sont celles de *portal.url*, *portal.user* et *portal.password*. Elles correspondent respectivement à l'URL de la base de données, au nom d'utilisateur et au mot de passe du serveur de base de données.

La ligne *portal.url* doit être modifiée de la façon suivante :

```
portal.url :{SGBD}://{Host}/{nom_de_la_base_de_donnees}?
autoReconnect=true&useUnicode=yes&characterEncoding=utf8
portal.user={nom d'utilisateur de la base de données}
portal.password={mot de passe d'accès à la base de données}
```

INITIALISER LA BASE DE DONNÉES

L'initialisation de la base de données se fait en exécutant à l'aide d'un terminal, la commande [ant](#) sur le fichier *WEB-INF/sql/build.xml* de la webapp.

La commande est la suivante :

```
ant
```

DÉPLOYER L'APPLICATION WEB

Après avoir configuré et initialisé la base de données, il suffit de démarrer le serveur d'applications.

INTÉGRER UNE CHARTE GRAPHIQUE

- 7. INTRODUCTION À L'INTÉGRATION GRAPHIQUE**
- 8. LES COMPOSANTS UTILISÉS**
- 9. LES PRINCIPAUX FICHIERS**
- 10. L'APPARENCE DES MENUS DU SITE**
- 11. LES STYLES DE RUBRIQUE**
- 12. LES THÈMES**

7. INTRODUCTION À L'INTÉGRATION GRAPHIQUE

Si **Lutece** est livré avec des thèmes, déclinaisons graphiques d'une charte, le cas le plus fréquent est l'intégration d'une charte graphique particulière qui aura été définie au préalable.

Intégrer une charte graphique dans un site **Lutece** revêt alors plusieurs aspects :

- A partir des livrables -images, css, js- intégrer ces éléments dans la structure du site Lutece
- Mettre en place une structure graphique spécifique en préparant les modèles de pages HTML (templates) et de rubriques (XSL)
- Préparer les déclinaisons de charte dans des thèmes

Le travail principal consistera à modifier des fichiers HTML, CSS, javascript et XSL pour :

- adapter l'architecture générale du site,
- définir des bandeaux ou pieds de page,
- modifier les menus,
- structurer le contenu des pages.

La plupart des technologies utilisées sont des standards couramment mis en œuvre et il ne s'agira ici que de comprendre comment **Lutece** les exploite, mais il sera possible d'aller plus loin en ayant recours à des éléments particuliers et plus rares.

8. LES COMPOSANTS UTILISÉS

Lutece embarque [Twitter Bootstrap](#) depuis plusieurs années, et inclut actuellement la version 3 du framework afin de bénéficier des fonctionnalités responsive et de faciliter l'intégration des divers modèles HTML du projet.



Chaque modèle de page utilise la grille Bootstrap 3 permettant ainsi de bénéficier rapidement des mises en pages par défaut et de leurs fonctionnalités responsive. Depuis la version 5 de **Lutece**, en complément des webfonts de Bootstrap 3, la bibliothèque d'icônes vectoriels [Font Awesome](#) a été intégrée dans les bibliothèques par défaut.



La bibliothèque javascript utilisée par défaut dans le projet **Lutece** est [jQuery](#) dans sa version 11. Certaines fonctionnalités javascript sont assurées par les fonctions incluses dans les librairies javascript fournies par Twitter Bootstrap.



L'ensemble des données passées dans les pages html utilisées dans **Lutece** sont fournies par l'utilisation de la bibliothèque [Freemarker](#) en utilisant le FreeMarker Template Language (FTL).

Pour la mise en forme des rubriques de contenus insérées dans les pages, la technologie [XSLT](#) est utilisée afin de pouvoir aisément changer le rendu des contenus par l'application de feuilles de style.

9. LES PRINCIPAUX FICHIERS

Pour se lancer dans l'intégration graphique d'un site généré avec **Lutece**, il faut surcharger, modifier les éléments fournis par l'outil, voir ajouter des fichiers ou des bibliothèques css/js.

Dans tous les cas l'intégration d'une charte graphique nécessitera la surcharge des modèles html, des feuilles XSL pour la partie document, de la feuille de style css principale *page_template_styles.css*.

Les éléments de charte d'un site Lutece se situent dans l'arborescence suivante

NB: cet exemple inclut le plug-in "document".

- css/
 - page_template_styles.css : Fichier css principal à modifier et surcharger;
 - plugins/
 - document/
 - document.css : Fichier contenant des surcharges nécessaires au plug-in "document".
- images/
 - local/
 - skin/
 - plugins/
 - document/
- themes/
 - theme1/
 - css/
 - page_template_styles.css : Fichier css décliné pour le thème choisi.
 - js/
 - images/
- WEB-INF
 - templates
 - skin
 - plugins
 - document/
 - document_content_service.html : Modèle d'affichage du contenu d'un document.
 - search/
 - search_results.html : Modèle d'affichage des résultats de recherche.
 - sites/
 - page_frameset.html : Modèle principal qui englobe l'ensemble des modèles de pages.
 - page_header_home.html : Modèle de l'entête de page de la page d'accueil.
 - page_header_internal.html : Modèle de l'entête de page des pages internes.
 - page_template_[1..n].html : Modèle de page avec diverses présentations: 1 colonne, 2 colonnes...
 - page_footer.html : Modèle du pied de page.

Pour intégrer la charte souhaitée, il faut de façon presque systématique intervenir sur les fichiers suivants qui vont contenir l'ensemble du code statique et généré par l'outil.

LE CONTENEUR DE PAGES : LE FICHIER PAGE_FRAMESET.HTML

Le fichier *page_frameset.html* est le fichier conteneur de pages qui intègre les [bibliothèques par défaut](#), le thème par défaut, les surcharges css et js fournis par les plugins du site ainsi que les balises *freemarker* qui permettent l'inclusion des modèles *header*, *footer*, *include* et des *modèles de page* qui seront décrits ci-après.

Le fichier se présente comme suit:

```
<!DOCTYPE html>
<html lang="fr">
<head>
<base href="{base_url}">
<title>{favourite} - {page_name}</title>
<meta http-equiv="x-ua-compatible" content="IE=edge" >
<meta charset="{encoding}">
...
<!-- Set the viewport width to device width for mobile -->
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Le style -->
<#if plugin_theme?? >
  <#assign thetheme=plugin_theme />
<#else>
  <#assign thetheme=theme />
</#if>
<link href="{thetheme.pathCss}/bootstrap.min.css" rel="stylesheet">
<link href="{thetheme.pathCss}/bootstrap-theme.min.css" rel="stylesheet">
<link href="{thetheme.pathCss}/page_template_styles.css" rel="stylesheet">
...
${plugins_css_links}
<!-- Included JS Files -->
<script src="js/jquery/jquery.min.js"></script>
</head>
<body>
  <!-- header -->
  <header role="banner">
    ${page_header}
  </header>
  <!-- end header -->
```

```

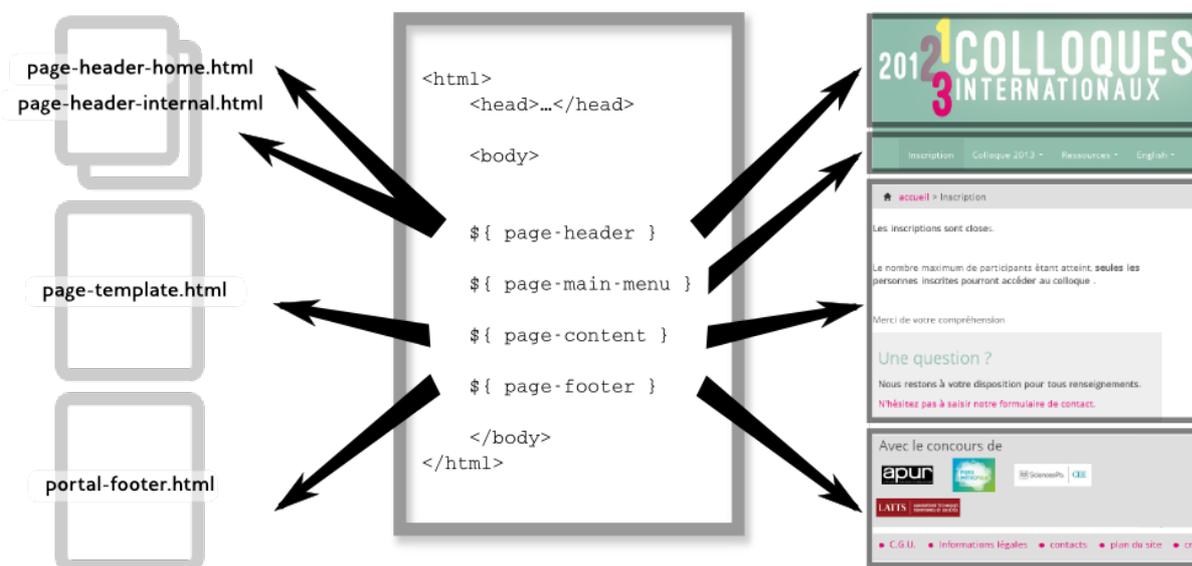
<div class="container">
<!-- container -->
${page_main_menu}
${page_path}
${page_content}
<!-- container -->
</div>
<footer>
${page_footer}

${include_1!}

</footer>
<!-- Le javascript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="js/bootstrap.min.js"></script>
${plugins_javascript_links}
</body>
</html>

```

Les balises ou bookmark dans le langage Freemarker, délimité par "\${...}", permettent de substituer du code produit par **Lutece** lors de la génération de la page du site.



Détaillons les balises les plus significatives:

- **`\${page_header}`** insère le contenu du template `page_header_home.html` pour la page d'accueil ou `page_header_internal.html` pour toutes les autres pages ;
- **`\${page_main_menu}`** permet d'insérer le menu du site;
- **`\${page_path}`** insère le fil d'ariane sur les pages internes du site ;
- **`\${page_content}`** insère le contenu du template de page défini dans l'administration du site `page_template_[1..n].html` pour la page ;
- **`\${page_footer}`** insère le contenu du template `page_footer.html` ;
- **`\${include_1!}`** insère un include soit un morceau de code HTML géré de façon spécifique dans **Lutece**.

LES MODÈLES DE PAGE HTML

Les balises ci-dessus permettent d'intégrer le contenu des fichiers .html ci-dessous. Lors de l'appel de l'url de la page, Lutece agrège l'ensemble du contenu de ces fichiers, selon l'ordre défini par l'emplacement des balises, et produit le contenu HTML final dans le navigateur Web.

LES FICHIERS PAGE_HEADER_HOME.HTML ET PAGE_HEADER_INTERNAL.HTML

Ils insèrent le code html correspondant à l'entête du site par le `page_header_home.html` pour la page d'accueil ou par le `page_header_internal.html` pour toutes les autres pages.

LE BOOKMARK `\${PAGE_MAIN_MENU}`

Ce bookmark insère le [menu](#) qui est géré par défaut en XSL ou, si le plugin menu est installé, par des modèles de page spécifiques (voir chapitre [Gérer l'apparence des Menus](#)).

LE FICHIER PAGE_PATH.HTML

Ce fichier permet d'insérer le code html correspondant au fil d'ariane. Le modèle contient le bookmark ``${page_path}`` qui est généré par la xsl `page_path.xsl` qui se trouve dans la gestion des feuilles de styles du site -menu *Charte > Gestion des feuilles de style XSL*.

LES FICHIERS PAGE_TEMPLATE_[1..N].HTML

Il est possible de disposer de plusieurs modèles de page dans un même site. Vous pouvez créer autant de modèles que désiré afin de guider le contributeur dans le positionnement des rubriques.

Lutece utilisant le framework CSS Twitter Bootstrap, l'intégration css doit utiliser les classes css de gestion des grilles apportées par le framework pour gérer la mise en page.

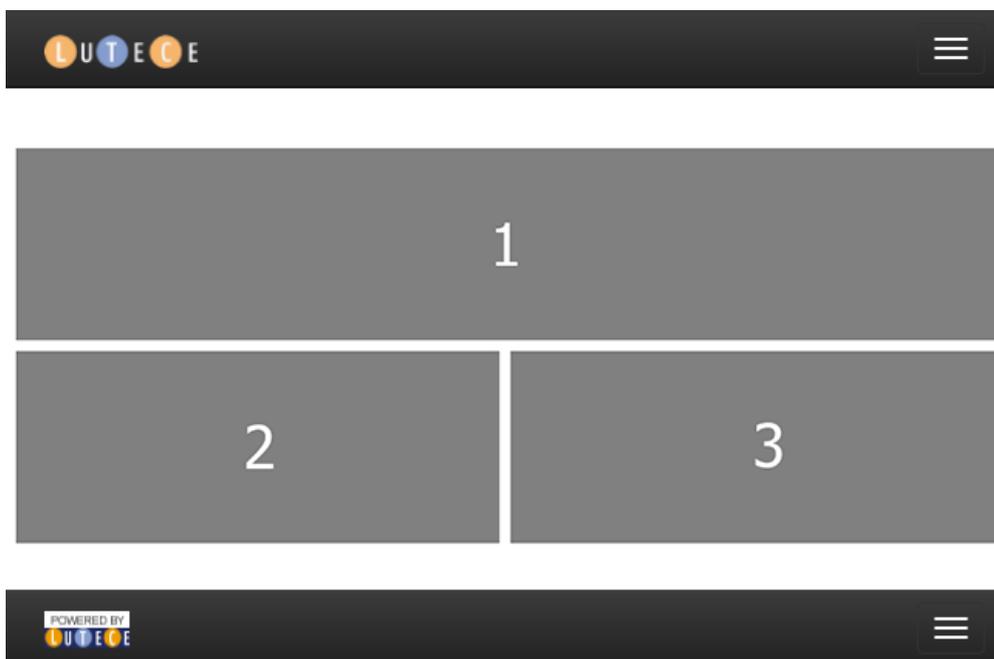
Il est conseillé de prendre en compte le rendu en mode "responsive design" pour permettre une vue adaptative de la page selon la taille de l'écran

Exemple de fichier:

```
<div class="row">
  <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
    `${page_content_col1}`
  </div>
</div>

<div class="row">
  <div class="col-xs-12 col-sm-12 col-md-6 col-lg-6">
    `${page_content_col2}`
  </div>
  <div class="col-xs-12 col-sm-12 col-md-6 col-lg-6">
    `${page_content_col3}`
  </div>
</div>
```

Dans l'exemple ci-dessus, le modèle présente une page avec 3 colonnes, la première ``${page_content_col1}`` fait 100% de la largeur de la page, et les deux colonnes suivantes font chacune 50% de la largeur de la page, pour un positionnement des colonnes qui se fera de la façon suivante :



Une fois les modèles créés, il faudra saisir dans la fonctionnalité *Modèles de page* du menu *Charte* du back-office, les informations nécessaires à l'utilisation des modèles définis dans la fonctionnalité d'Administration du site.

LE FICHIER PAGE_FOOTER.HTML

Ce fichier permet d'insérer le code html correspondant au pied de page du site.

LES INCLUSIONS DE FICHIER OU PAGE INCLUDE

Certains plugins peuvent proposer des *page include* qui vont permettre d'inclure du code HTML dans les pages du site par le biais de **bookmarks freemarker** placés dans le gabarit principal du portail, dans l'exemple du `page_frameset.html` le ``${include_1!}``.

LES FICHIERS CSS

LE FICHIER PAGE_TEMPLATE_STYLE.CSS

C'est le fichier css principal qui contient toutes les surcharges nécessaires à l'application de la charte du site. Il doit évidemment être le plus petit et complet possible afin de couvrir un maximum de fonctionnalités du site (pages, formulaires...) et éviter ainsi les surcharges trop importantes dans les css des plugins.

LES FICHIERS CSS DES PLUGINS

Certains plugins ont par défaut une feuille de style qui se trouvera dans l'arborescence du site dans `css/plugins/nom_du_plugin/nom_du_plugin.css`.

Ces fichiers css apparaîtront après substitution du tag `#{plugins_css_links}` dans le *header* du `page_frameset.html` lors de la génération de la page.

LES FICHIERS JAVASCRIPT

En sus des bibliothèques définies par défaut dans *Jquery* et des fichiers css apportés par *Bootstrap*, vous pouvez ajouter notamment dans le fichier `page_frameset.html` votre(vos) propre(s) bibliothèque(s) javascript, que vous placerez dans le dossier `/js` situé à la racine du site ou dans le répertoire `"theme\themeXX\js"` s'il s'agit d'un fichier js concernant une déclinaison.

LES FICHIERS JS DES PLUGINS

Certains plugins ont par défaut des bibliothèques javascript qui se trouvent dans l'arborescence du site dans `js/plugins/nom_du_plugin/*.js`.

Les fichiers javascript qui sont inclus par les plugins apparaîtront après substitution du tag `#{plugins_javascript_links}` en fin du `page_frameset.html` lors de la génération de la page.

10. L'APPARENCE DES MENUS DU SITE

Lutece offre deux possibilités pour le choix du menu : utiliser le menu issu du coeur (géré par une feuille de style XSL : menu_main.xml) ou le plugin Menus pour une personnalisation plus avancée.

LE MENU PAR DÉFAUT

Nativement, le menu d'origine du site est constitué par la feuille de style XSL qui se trouve dans *Charte > Gestion des feuilles de style XSL* sous le nom de "Menu principal".

Il permet d'intégrer un menu simple qui affichera les pages de premier niveau.

Exemple d'insertion dans le fichier *page_frameset.html* :

```
div class="container">
<!-- container -->
${page_main_menu}
${page_path}
${page_content}
<!-- container -->
</div>
```

Il existe deux autres feuilles de style qui permettent d'afficher les pages sous forme de liste et de sous-liste : *menu_init.xml* ou sous forme d'arborescence : *menu_tree.xml*



Pour obtenir le rendu de ce menu, il faut insérer dans le fichier conteneur *page_frameset.html* la balise :

```
${page_init_menu}
```

UTILISER LE PLUGIN MENUS

Le plugin Menus permet de concevoir des menus particuliers qui vont s'insérer dans le site par des bookmarks. La personnalisation de ces menus se fait dans les templates html se trouvant dans *WEB-INF/templates/skin/plugins/menus*.





MENU CLASSIQUE UN NIVEAU

Le fichier *main_tree_pages_list.html* affiche le premier niveau d'arborescence du menu :

```
<#if menu.childs?has_content>
  <div id="content">
    <ul>
      <#list menu.childs as submenu>
        <li><a href="{site_path}?page_id={submenu.page.id}"
target="_top">{submenu.page.name} - {submenu.page.description}</a></li>
      </#list>
    </ul>
  </div>
</#if>
```

Ce template de menu peut être inclus dans *page_frameset.html* via la balise Freemarker suivante :

```
{page_tree_menu_main_all_pages}
```

MENU CLASSIQUE 2 NIVEAUX

Le fichier *main_tree_pages_list_tree.html* affiche quant à lui le premier niveau d'arborescence ainsi que le niveau inférieur :

```
<#if menu.childs?has_content>
  <div id="content">
    <ul>
      <#list menu.childs as submenu >
        <a href="{site_path}?page_id={submenu.page.id}" target="_top">{submenu.page.name} -
{submenu.page.description}</a></li>
        <#if submenu.childs?has_content>
          <ul>
            <#list submenu.childs as subsubmenu >
              <li><a href="{site_path}?page_id={subsubmenu.page.id}"
target="_top">{subsubmenu.page.name} - {subsubmenu.page.description}</a></li>
            </#list>
          </ul>
        </#if>
      </#list>
    </ul>
  </div>
</#if>
```

Ce template de menu peut être inclus dans *page_frameset.html* via la balise Freemarker suivante :

```
{page_tree_menu_tree_all_pages}
```

MENU XPAGES

Dans le cadre de développements spécifiques, les contenus sont affichés via des pages dites "XPages", pour lesquelles il existe également un type de menu qui affiche exclusivement les XPages, et dont le template est le fichier *xpages_list.html* :

```
<ul>
  <#list xpages_list as xpage>
    <li><a href="{site_path}?page={xpage.name}">{xpage.name}</a></li>
  </#list>
</ul>
```

Ce template de menu peut être inclus dans *page_frameset.html* via la balise Freemarker suivante :

```
{xpage_menu}
```

11. LES STYLES DE RUBRIQUE

Les contenus des rubriques (portlet) sont mis en forme via des styles, gérés dans des feuilles de styles XSL. Un type de rubrique peut afficher un même contenu de différentes manières en changeant simplement de style dans l'administration du site.

Les styles des portlets et des documents se créent en deux étapes : le nommage du style puis l'ajout de la feuille de style correspondante.

AJOUT D'UN NOUVEAU STYLE

Dans *Charte > Gestion des styles*, créer un nouveau style.

Création d'un style

Identifiant *	<input type="text" value="1740"/>
Nom *	<input type="text" value="Rubrique Document Actualité"/>
Type de composant	<input type="text" value="Rubrique"/>
Type de rubrique	<input type="text" value="Document"/>

Choisir un type de composant (qui sera le plus communément une rubrique), le numéroter manuellement de façon à le grouper avec les autres éléments du même type et le nommer de façon explicite, pour le retrouver plus facilement.

Sélectionner ensuite le type de rubrique si le composant est une rubrique : rubrique de type document, rubrique de type HTML, rubrique de type liste de documents, etc.

CRÉATION D'UNE FEUILLE DE STYLE XSL ET L'ASSOCIER AU STYLE

Dans *Charte > Gestion des feuilles de style XSL*, cliquer sur le bouton *Ajouter une feuille de style XSL*.

Ajouter une feuille de style XSL

Nom de la feuille de style : *	<input type="text" value="Rubrique Document Actualité"/>
Fichier : *	<input type="text" value=""/> <input type="button" value="Parcourir"/>
Style : *	<input type="text" value="Rubrique Document - Rubrique Document Actualité"/>
Mode : *	<input type="text" value="Normal"/>

Nommer la feuille de style de la même façon que le style précédemment créé de façon à faire facilement la liaison entre les deux.

Dans la liste déroulante *Style*, sélectionner le style précédemment créé.

Dans la liste déroulante *Mode*, sélectionner "Normal".

Note : Il est possible de reprendre une feuille de style XSL incluse par défaut et de l'adapter au portlet ou document.

12. LES THÈMES

Les thèmes permettent de proposer aux utilisateurs du site une déclinaison de la charte graphique principale. Il s'agit en effet d'apporter des modifications de couleurs, d'images mais pas de la structure du site.

Certains thèmes sont livrés avec **Lutece**, d'autres peuvent être ajoutés ou créés.

Le menu *Charte > Gestion des thèmes* du back-office permet d'avoir un aperçu des thèmes disponibles.

The screenshot displays the 'Gestion des thèmes' (Theme Management) interface in the Lutece back-office. The top navigation bar includes 'Contenu', 'Site', 'Applications', 'Utilisateurs', 'Gestionnaires', 'Charte', and 'Système'. The user is logged in as 'admin Admin'. The interface shows a list of themes, each with a preview and metadata.

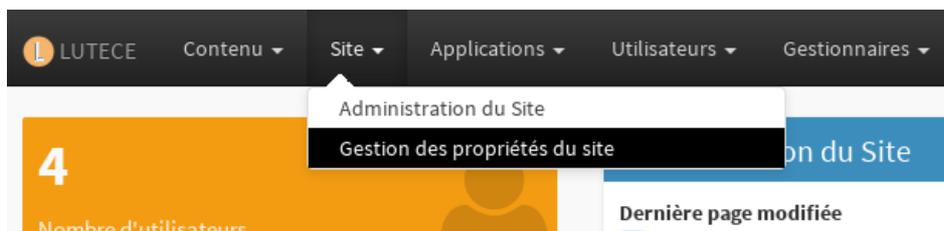
- Thème Défait**: Marked as 'Thème global sélectionné'. Metadata: Auteur: Mairie de Paris, Version: 1.0, Licence: BSD. Preview shows a Paris-themed page with the text 'Qui utilise Lutèce?' and 'PARIS'.
- Amelia**: Metadata: Auteur: Mairie de Paris, Version: 1.0, Licence: Apache 2.0. Preview shows a teal-themed page with the text 'Amelia Sweet and cheery' and a navigation bar with 'Default', 'Primary', 'Success', 'Info', 'Warning', and 'Danger' buttons.
- Cosmo**: Metadata: Auteur: Mairie de Paris, Version: 1.0, Licence: Apache 2.0. Preview shows a white-themed page with the text 'Cosmo An ode to Metro' and a navigation bar with 'Default', 'Primary', 'Success', 'Info', and 'Warning' buttons.
- Cyborg**: Metadata: Auteur: Mairie de Paris, Version: 1.0, Licence: Apache 2.0. Preview shows a dark-themed page with the text 'Cyborg'.

CONFIGURER LES FONCTIONNALITÉS AVANCÉES

- 13. GÉRER LES PROPRIÉTÉS DU SITE
- 14. GÉRER LES TYPES DE DOCUMENTS
- 15. GÉRER LES LIBELLÉS
- 16. PARAMÉTRER LES EXTENSIONS
- 17. GÉRER L'AUTHENTIFICATION DU FRONT OFFICE
- 18. GÉRER LES UTILISATEURS BACK-OFFICE
- 19. GÉRER LES DROITS DES UTILISATEURS
- 20. GÉRER LES RÔLES DES UTILISATEURS
- 21. GROUPES DE TRAVAIL
- 22. DÉFINIR UN WORKFLOW
- 23. PARAMÉTRER UN PROCESSUS DE GESTION DE DEMANDE

13. GÉRER LES PROPRIÉTÉS DU SITE

Nous conseillons de commencer par le contrôle des propriétés du site à partir du menu *Site > Gestion des propriétés du site*.



Une page contenant un nombre assez important de champs de texte va se charger. Ces champs sont généralement remplis avec des textes par défaut. Ainsi le nom du site contient le mot **Lutece** lors de l'installation initiale. Il convient donc de changer sa valeur pour qu'elle identifie au mieux votre site.

Modification des propriétés du site

Propriétés par défaut du site

Ces propriétés vous permettent de définir le nom du site, les valeurs des tags méta par défaut, et plus encore.

Nom du site

Ce nom s'affiche dans le titre des pages et le bandeau du back office

Tag meta - Auteur

Saisir l'auteur du site

Tag meta - Copyright

Saisir les informations concernant les droits d'auteur

Tag meta - Description

Après modification, il ne faut pas oublier de cliquer sur le bouton "Modifier" en bas de page.

Il convient de la même façon de vérifier tous les champs. Les changements auront des impacts différents selon leur rôle. Dans le cas du titre par exemple, on peut observer le changement du nom du site dans le back-office, dans l'onglet qui affiche le site, dans la barre de titre du navigateur, et (invisible ici) dans l'historique de navigation pour permettre aux visiteurs de la retrouver plus facilement.

Vous pourrez modifier :

- la **description** meta, connue pour intervenir dans l'affichage des résumés de page dans les moteurs de recherche
- les **mots-clés** à associer par défaut au site
- les **mentions de copyright** qui doivent être mis en méta (informations invisibles à l'utilisateur mais utiles aux outils d'indexation), au niveau des mentions légales (ce texte apparaît le plus souvent en bas de page)
- les **e-mails** associés au site, e-mail du webmaster qui sera le contact par défaut des formulaires, e-mail no-reply souvent associé à des fonctions particulières...

Si certaines informations ne sont pas en votre possession et que vous n'êtes pas en mesure d'obtenir l'information, ne les changez pas.

14. GÉRER LES TYPES DE DOCUMENTS

Un document (du plugin Document), est défini par son type, c'est-à-dire par les informations qui le structurent. Un document possède par défaut un titre et une description. Le type va être spécialisé par l'ensemble des autres champs (adresse, fichier, image...) qui le décriront.

Par défaut, le plugin Document propose des types de documents prédéfinis, à savoir :

- Article
- Bannière
- Brèves
- Fiches acteurs
- Fichiers en téléchargements
- Images
- Vidéos en ligne

Chacun des types est composé par un ou plusieurs éléments, ou type de champs, que l'on peut utiliser pour le définir :

- Date
- Fichier
- Image
- Lien interne
- Liste de valeurs
- Texte multiligne
- Texte numérique
- Texte riche
- Texte
- Url

Ce paramétrage peut être modifié en fonction des besoins. Il est possible de créer des types de documents particuliers pour se conformer au plus près aux besoins éditoriaux. Il est possible par exemple de créer un type de document "Projet" qui permettra de décrire des projets en y ajoutant des contenus éditoriaux, des images, un fichier PDF et une URL.

CRÉER UN TYPE DE DOCUMENT

Par le menu *Contenu > Gestion des types de documents*, vous pouvez accéder au bouton de création d'un nouveau type de document. Le code du document servira à l'appeler dans les feuilles de style XSL et ne pourra pas être modifié par la suite.

The screenshot shows a web interface titled "Gestion des types de documents" with a subtitle "Création d'un nouveau type de document". The main section is titled "Propriétés" and contains four input fields:

- Code (nom interne)***: A text input field containing "projet". Below it, the text reads: "Exemple : 'article' (saisie en minuscules et sans espaces) Non modifiable par la suite".
- Nom***: A text input field containing "Projet Test". Below it, the text reads: "Exemple : 'Article'".
- Description***: A text input field containing "Document de type Projet (test)". Below it, the text reads: "Exemple : 'Document de type article'".
- Modèle de métadonnées***: A dropdown menu with "None" selected. Below it, the text reads: "Sélectionner un modèle".

At the bottom of the form, there are two buttons: "✓ Enregistrer et ajouter les attributs" (highlighted in blue) and "⊗ Annuler".

Une fois les informations saisies, vous pouvez gérer les attributs du nouveau type de document.

N.B. : Le modèle de métadonnées défini par défaut est "none" (soit pas de modèle défini) sachant qu'est disponible aussi le modèle "[Dublin-core](#)", un schéma de métadonnées générique qui permet de décrire des ressources numériques ou physiques et d'établir des relations avec d'autres ressources.

LES PROPRIÉTÉS D'UN TYPE DE DOCUMENT

Après la création du document, il est possible de modifier toutes les informations saisies précédemment à l'exception du code du type de document.

CRÉER DES ATTRIBUTS

Pour ajouter un attribut, il faut choisir son type parmi les valeurs proposées dans la liste déroulante (date, fichier, texte, image...) et cliquer sur "Ajouter".

On retrouve des attributs communs à tous les champs, titre, code interne, description, déterminer si le champ est obligatoire

en saisie, s'il doit être indexé par le moteur de recherche, et, en fonction des types, des attributs particuliers :

- Liste des extensions acceptées pour les attributs fichier et image
- Longueur du champ et nombre de caractères maximum acceptés pour l'attribut texte et texte numérique
- Définition de la date par défaut pour l'attribut date
- Liste des valeurs et valeur par défaut pour l'attribut liste de valeur.

N.B.: Si le plugin "plugin-regularexpression" permettant de gérer les expressions régulières est installé, vous pouvez appliquer à chaque attribut des règles contrôlées par des expressions régulières, afin de vérifier la valeur saisie par l'utilisateur. Par défaut, une règle de vérification d'e-mail existe, qui associée à l'attribut "E-mail" vérifiera la conformité d'adresse e-mail saisie. Vous pouvez ajouter autant de règles que nécessaire.

Gestion des types de documents Ajout d'un attribut à un type de document

Ajout d'un attribut de type :

Nom*	<input type="text"/>
	Exemple : 'Titre'
Code (nom interne)*	<input type="text"/>
	Exemple : 'title' (accents et espaces interdits)
Description*	<input type="text"/>
	Exemple : 'Champs titre'
Obligatoire	<input type="checkbox"/> Saisie du champ obligatoire
Indexé	<input type="checkbox"/> Champ indexé par le moteur de recherche
	<input type="button" value="+ Ajouter"/> <input type="button" value="⊗ Annuler"/>

Une fois les attributs créés ils apparaissent dans la liste :

Gestion des types de documents Modification d'un type de document

Propriétés

Code (nom interne)* :

projet

Nom* :

Projet Test

Description* :

Document de type Projet

Url de l'icône par défaut :

Attribut à utiliser comme icône :

Aucun attribut

Modèle de métadonnées* :

None

Sélectionner un modèle

✓ Modifier

⊗ Annuler

Liste des attributs

Ajouter un attribut de type :

Nom de l'attribut	Code	Type	Obligatoire	Indexé	Actions
Illustration Illustration du projet	illustration	image	Non	Non	   
Description du projet Description du projet	description	richtext	Non	Non	   
Coût du projet Coût du projet en Euros	cout	numerictext	Non	Non	   

Feuilles de style XSL

Back office

 Parcourir

Mettre à jour le fichier

Aucun fichier

Front office

 Parcourir

Mettre à jour le fichier

Aucun fichier

Nous avons donc créé un type de document "projet" avec trois attributs : illustration, description et coût du projet.

CRÉATION D'UN DOCUMENT

L'interface de création des documents se trouve dans *Contenu > Gestion des documents*.

Afin d'être en mesure de créer un nouveau document, il doit exister un espace de documents acceptant le nouveau type de document créé. A cet effet, il est possible de :

- Créer un nouvel espace de documents en cliquant sur le bouton 
- Ou modifier un espace de documents existant en cliquant sur le bouton 

Dans la section "type de documents autorisés" de l'espace, cocher la case correspondant au nouveau type de document précédemment créé, et en veiller à ce que la case "Création de documents" soit cochée.

Gestion des documents Modification d'un espace

Modification d'un espace

Nom de l'espace* : Projets

Description* : Projets test

Affichage par défaut des documents : Détail

Icône :

Types de documents autorisés :

- Fiche acteur
- Article
- Brèves
- Image
- Video
- PDF
- Enregistrement sonore
- Bannières
- Flash
- Projet Test

<Non assigné à un groupe>

Création de documents : Création de documents
Permettre la création de document dans cet espace

Procéder ensuite à la création du document.

Après l'avoir validé, il faut ensuite le publier sur le site, par exemple dans une nouvelle rubrique de type "Document" via *Site > Administration du site*, en sélectionnant "Projet Test" dans le type de documents, puis en choisissant le style approprié créé au préalable. Il reste ensuite à publier le document dans la rubrique.

PERSONNALISER LE RENDU DU DOCUMENT EN FRONT OFFICE DANS UNE RUBRIQUE (PORTLET)

Pour un affichage du nouveau type de document créé dans un portlet, il faut passer par le menu *Charte > Gestion des Styles* puis *Charte > Gestion des Feuilles de style XSL* (voir le chapitre : [Gestion des styles](#) qui détaille la procédure de création d'un style).

La feuille de style XSL doit appeler les attributs du document créé au préalable dans *Contenu > Gestion des types de documents*.

Chacun des attributs du document peut être affiché via l'arbre XML :

```
document-portlet/document/document-xml-content/nom_document/nom_document-nom_attribut
```

pour un portlet document ;

```
document-list-portlet/document/document-xml-content/nom_document/nom_document-nom_attribut
```

pour un portlet liste de documents.

Note : Il est possible de reprendre une feuille de style XSL incluse par défaut et de l'adapter au nouveau document créé.

Voici une feuille de style XSL pour l'affichage du type de document "Projet" dans un portlet :

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:param name="site-path" select="site-path" />
<xsl:variable name="portlet-id" select="portlet/portlet-id" />

<xsl:template match="portlet">

  <xsl:variable name="device_class">
  <xsl:choose>
    <xsl:when test="string(display-on-small-device)='0'">hidden-xs</xsl:when>
    <xsl:when test="string(display-on-normal-device)='0'">hidden-sm</xsl:when>
    <xsl:when test="string(display-on-large-device)='0'">hidden-md</xsl:when>
    <xsl:when test="string(display-on-xlarge-device)='0'">hidden-lg</xsl:when>
    <xsl:otherwise></xsl:otherwise>
  </xsl:choose>
  </xsl:variable>
  <div class="{\$device_class}">
    <div class="row">
```

```

        <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
        <xsl:if test="not(string(display-portlet-title)='1')">
            <h2>
                <xsl:value-of disable-output-escaping="yes" select="portlet-name" />
            </h2>
        </xsl:if>
        </div>
    </div>
    <xsl:apply-templates select="document-portlet" />
</div>
</xsl:template>

<xsl:template match="document-portlet">
    <div class="row">
        <xsl:apply-templates select="document" />
    </div>
</xsl:template>

<xsl:template match="document">
    <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
        <xsl:apply-templates select="document-xml-content/projet" />
    </div>
</xsl:template>

<xsl:template match="document-xml-content/projet">
    <div class="">
        <div>
            <p>
                <a href="{\$site-path}?document_id={document-id}&#38;portlet_id={\$portlet-id}"
target="_top">
                    Titre du projet : <xsl:value-of select="document-title" />
                </a>
            </p>
            <p><xsl:value-of disable-output-escaping="yes" select="substring(projet-
description,1,100)"/>...</p>
            <p><xsl:value-of select="projet-cout" /> €</p>
        </div>

        <div class="">
            <a href="{\$site-path}?document_id={document-id}&#38;portlet_id={\$portlet-id}"
target="_top">
                <xsl:choose>
                    <xsl:when test="projet-illustration/file-resource!=''">
                        <xsl:apply-templates select="projet-illustration/file-resource" />
                    </xsl:when>
                    <xsl:otherwise></xsl:otherwise>
                </xsl:choose>
            </a>
        </div>
    </div>
</xsl:template>

<xsl:template match="file-resource">
    <xsl:choose>
        <xsl:when test="(resource-content-type='image/jpeg' or resource-content-type='image/jpg' or
resource-content-type='image/pjpeg' or resource-content-type='image/gif' or resource-content-
type='image/png')">
            
        </xsl:when>
        <xsl:otherwise>
            <a href="document?id={resource-document-id}&id_attribute={resource-attribute-id}"
class="img-responsive" />
                
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

</xsl:stylesheet>

```

Voici un exemple de rendu simple, de notre document de type "Projet Test" dans la colonne 2 de la page d'accueil du site :

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce maximus auctor placerat. Integer id luctus nunc. Donec dignissim mauris tellus, in dapibus ante interdum quis. Cras quis ligula nec nisi interdum fringilla. Donec mollis diam et lectus volutpat malesuada pharetra sit amet libero. Praesent purus turpis, auctor sit amet dictum eget, imperdiet vel ante. Nullam a massa est. Sed a bibendum magna. Praesent ultricies, orci quis ultrices ultrices, mauris nisl cursus risus, ac euismod magna purus sed diam. Pellentesque a magna a est maximus hendrerit. Mauris a iaculis dui. Fusce ut metus ut turpis ullamcorper fringilla non facilisis dolor. Mauris lacinia tempus urna, nec mattis diam molestie nec. Sed ornare nisl non aliquet sagittis. Aliquam eleifend rhoncus luctus.

Rubrique Document Projet

Titre du projet : [Projet test](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque ut tempor enim. Sed tempus orci ...

350000 €



Le portlet document reprend le contenu des attributs présents dans la feuille de style XSL, et l'URL du titre renvoie vers l'affichage du document en pleine page.

PERSONNALISER LE RENDU DU DOCUMENT EN FRONT OFFICE EN PLEINE PAGE ("DOCUMENT CONTENT SERVICE")

Chaque document peut être affiché individuellement sur une page dédiée utilisant le "Content Service" du plugin Document, via le fichier `document_content_service_default_xsl.html`.

Pour personnaliser le rendu du document, il faut créer la feuille de style XSL pour chaque type de document, via *Contenu > Gestion des types de documents*, puis sélectionner le type de document dont on veut modifier la feuille de style. Il est possible d'ajouter ou de télécharger une feuille de style, pour le front office et le back office. Dans le cas de l'ajout ou de la mise à jour du fichier, il est nécessaire de cocher la case "Mettre à jour le fichier" pour que les changements soient pris en compte.

Feuilles de style XSL

<p>Back office</p> <div style="border: 1px solid #ccc; height: 20px; width: 100%;"></div> <input type="button" value="Parcourir"/> <p><input type="checkbox"/> Mettre à jour le fichier</p> <input type="button" value="Aucun fichier"/>	<p>Front office</p> <div style="border: 1px solid #ccc; height: 20px; width: 100%;"></div> <input type="button" value="Parcourir"/> <p><input type="checkbox"/> Mettre à jour le fichier</p> <input type="button" value="Télécharger la feuille de style XSL"/>
---	--

Vous pouvez récupérer une XSL existante en téléchargeant le fichier via le bouton "Télécharger la feuille de style XSL" sur l'un des types de documents par défaut (comme Article, par exemple). Il conviendra ensuite de la modifier pour qu'elle soit adaptée au nouveau type de document créé, en remplaçant le code du type de document par le code de celui que vous avez créé.

Voici un exemple de feuille de style XSL pour l'affichage pleine page d'un document de type "Projet Test" :

```
<?xml version="1.0"?>
```

```

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:param name="publication-date" select="publication-date" />
  <!--
    How to display the publication date :
    <xsl:value-of select="$publication-date" />
  -->
  <xsl:output method="html" indent="yes"/>

  <xsl:template match="content">
    <xsl:apply-templates select="projet"/>
  </xsl:template>

  <xsl:template match="projet">
    <div class="row">
      <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
        <h1><xsl:value-of select="document-title" /></h1>
      </div>
    </div>
    <div class="row">
      <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
        <xsl:choose>
          <xsl:when test="not(string(projet-illustration)='')">
            <xsl:apply-templates select="projet-illustration/file-resource" />
          </xsl:when>
          <xsl:otherwise></xsl:otherwise>
        </xsl:choose>
      </div>
      <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
        <xsl:choose>
          <xsl:when test="not(string(projet-description)='')">
            <xsl:value-of disable-output-escaping="yes" select="projet-description" />
          </xsl:when>
          <xsl:otherwise></xsl:otherwise>
        </xsl:choose>
      </div>
      <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
          <xsl:if test="not(string(projet-cout)='')">
            Coût du projet : <xsl:value-of select="projet-cout" />
          </xsl:if>
        </div>
      </div>
    </div>
  </xsl:template>

  <xsl:template match="file-resource">
    <xsl:choose>
      <xsl:when test="(resource-content-type='image/jpeg' or resource-content-type='image/jpg'
or resource-content-type='image/pjpeg' or resource-content-type='image/gif' or resource-content-
type='image/png')">
        
      </xsl:when>
      <xsl:otherwise></xsl:otherwise>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>

```

Il est possible de choisir parmi plusieurs **modèles de présentation** pour l'affichage pleine page du document. Ce choix s'effectue dans l'écran d'édition d'un document via *Contenu > Gestion des documents*, éditer le document dans l'espace de Documents auquel il est affecté, et dans *Propriétés avancées*, trois modèles de présentation sont sélectionnables :

- Défaut (correspondant au fichier *WEB-INF/templates/skin/plugins/document/document_content_service.html*)
- Pleine page (correspondant au fichier *WEB-INF/templates/skin/plugins/document/page_templates/page_template_document1.html*)
- Rubrique en-dessous (correspondant au fichier *WEB-INF/templates/skin/plugins/document/page_templates/page_template_document2.html*)

Propriétés avancées

Période de validité

Début

Fin

Modèle de présentation :

Défaut

Pleine page

Rubrique en-dessous

Choix du modèle de présentation du document sur le site

Ces fichiers sont pleinement personnalisables pour répondre au mieux à la charte graphique du site. Les templates par défaut incluent les *Extenders* (balises Freemarker utilisées pour les composants du plugin Extend - voir le chapitre [Paramétrer les Extensions](#)).

Voici un exemple de rendu simple du document affiché en pleine page via le template "pleine page" :

L U T E C E

Projet test



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque ut tempor enim. Sed tempus orci et arcu tincidunt, ac convallis velit volutpat. Quisque neque diam, vestibulum sed mollis eget, consequat eget mauris. Nam venenatis nisl congue, porta leo eu, viverra orci. Donec vel lacinia neque, bibendum elementum dui. Praesent vel tristique est. Mauris tempor ut augue vitae iaculis. Vivamus ullamcorper odio euismod enim tristique, in viverra purus finibus. Nam ultrices viverra mi, quis sollicitudin urna ultricies lobortis. Quisque urna lectus, tincidunt et est sed, dapibus semper elit.

Etiam sagittis turpis non tincidunt consectetur. Etiam vulputate convallis velit, nec tincidunt tellus facilisis vitae. Praesent id justo sit amet dui iaculis vehicula. Nulla tempus ligula leo, faucibus vestibulum sem hendrerit ut. Pellentesque laoreet, nisl a dictum sagittis, nulla augue eleifend enim, non semper metus metus ac ligula. Praesent id auctor nisl. Curabitur pharetra massa nec leo luctus mattis. Vestibulum at dictum ligula. Quisque commodo orci mi, laoreet maximus risus suscipit eget. Praesent sit amet eros blandit, iaculis nulla ut, finibus elit. Nam bibendum tempus augue sit amet pharetra. Nunc in diam velit. Vivamus eu risus suscipit, laoreet erat in, commodo nibh.

Coût du projet : 350000

Rubrique Document Projet

POWERED BY
L U T E C E
accueil
plan du site
contacts
crédits

Note : Un document doit obligatoirement être publié dans un portlet pour pouvoir être visualisé en front office. L'ID du portlet figure ainsi dans l'URL du document, exemple : `/jsp/site/Portal.jsp?document_id=24&portlet_id=3`

15. GÉRER LES LIBELLÉS

Le plug-in *plugin-sitelabels* permet d'afficher du contenu texte sur les pages du site en front office. Celui-ci est éditable grâce à des zones de texte dans le back office du site, donnant la main aux utilisateurs et webmasters pour modifier eux-mêmes certains contenus définis.

Rappel : il faut au préalable installer le plugin en l'ajoutant dans le POM du site, puis l'activer dans *Système > Gestion des plugins* du back office. Ensuite, dans *Système > Agencement des menus du back office*, affecter les fonctionnalités du plugin à l'un des Groupes du menu.

AJOUTER UN LIBELLÉ

Dans le back-office, sélectionner "*Gestion des libellés spécifiques*" dans le menu, puis sur "*Ajouter un libellé*". Nommer la clé en minuscules, et saisir sa valeur.

Note : Le champ de valeur peut être transformé en zone de texte étendue en ajoutant ".textblock" à la fin du nom de la clé.

Gestion des libellés

Clé * :

Saisir la clé du libellé (sans préfix). Ajouter le suffix "textblock" par obtenir un textarea.

Valeur du libellé * :

Valeur du libellé

La clé et sa valeur apparaissent désormais dans la liste des libellés utilisables sur les pages du site.

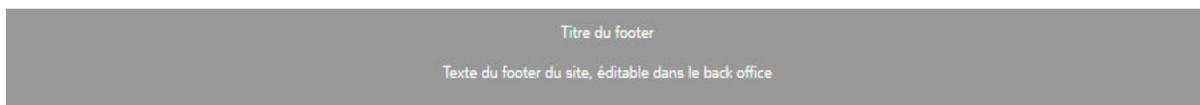
Clé	Valeur du libellé	Actions
sitelabels.site_property.footertitle	Titre du footer	 
sitelabels.site_property.footer.textblock	Texte du footer du site, éditable dans le back office	 

UTILISER UN LIBELLÉ DANS LES PAGES DU SITE

Dans le code HTML de la page, insérer `#dskey{sitelabels.site_property.clé}` à l'endroit où doit apparaître le texte.

Exemple : Sitelabel pour le footer du site.

```
<footer>
  <div class="col-xs-12 col-sm-12 col-md-6 col-lg-6 footer">
    <p>#dskey{sitelabels.site_property.footertitle}</p>
    <p>#dskey{sitelabels.site_property.footer.textblock}</p>
  </div>
</footer>
```



16. PARAMÉTRER LES EXTENSIONS

Partager ses contenus de façon aisée, voter, proposer de commenter, avec modération ou sans modération, les articles de son site en un click, c'est ce que permet le plug-in *plugin-extend*. Chacune des extensions possède une configuration par défaut qu'il est possible de paramétrer.

LE PARAMÉTRAGE PAR DÉFAUT DES EXTENSIONS

Le paramétrage est accessible depuis le menu *Contenu > Gestion des extensions de contenu* par le bouton "Modifier la configuration par défaut".

L'écran permet de modifier le paramétrage par défaut de l'ensemble des extensions en utilisant la liste des extensions et en cliquant sur le bouton "Modifier la configuration par défaut" pour passer d'une extension à une autre.

Les réseaux sociaux

Il faut dans cet écran choisir quels seront les réseaux sociaux affichés par défaut. Accéder au paragraphe ["Gérer les réseaux sociaux"](#)

The screenshot shows the configuration interface for the 'Réseaux sociaux' extension. At the top, the title 'Gestion des extensions de contenu' is followed by 'Configuration de l'extension' and an 'Aide' icon. Below this, the current extension is identified as 'Réseaux sociaux' with a dropdown arrow, and a 'Modifier la configuration par défaut' button is visible. The main section is titled 'Réseaux sociaux à afficher :'. It contains four checkboxes, each with a corresponding social media icon and label: Facebook, Google+, Twitter (share), and Twitter (follow). At the bottom of this section, there are two buttons: 'Modifier' (with a checkmark icon) and 'Annuler' (with an 'x' icon).

Les votes

Les options disponibles sont ici le choix d'une notification, du type de vote, de la restriction sur le vote: période d'activation, unicité, authentification et limite du nombre. Accéder au paragraphe ["Gérer les votes"](#)

Configuration par défaut de l'extension

Vote



Notification par email : <Pas de notification>

Sélectionner la liste de diffusion à notifier lorsqu'un vote est effectué.

Type de vote : Star

Période d'activation du vote : Du au

- Options**
- Vote unique (L'internaute ne pourra voter qu'une seule fois la ressource.)
 - Vote temporel (L'internaute pourra voter plusieurs fois la ressource dans le temps.)
 - Cochez si l'utilisateur doit être authentifié pour voter.
 - Cochez si l'utilisateur peut annuler son vote.

Limite de vote :

- Désactivé
- Activé

Les commentaires

Les options sont la modération, le nombre de commentaires à afficher, l'emplacement du formulaire de commentaires, l'utilisation d'un éditeur en front-office, les notifications et la gestion des messages.

Configuration par défaut de l'extension

Commentaires ▾

[Modifier la configuration par défaut](#)

Modération des commentaires : Oui Non

Nombre de commentaires à afficher :

Autoriser les sous-commentaires : Oui Non

Emplacement du formulaire de soumission de commentaire :

Utiliser un éditeur BBCode : Oui Non

Notification par email :

Sélectionner la liste de diffusion à notifier lorsqu'un commentaire est soumis.

Badge affiché sur les message des administrateurs :

Fichier ▾ Editer ▾ Insérer ▾ Voir ▾ Format ▾ Tableau ▾ Outils ▾

↶ ↷ 📄 Formats ▾ **B** *I* A ▾ A ▾ ☰ ☷ ☹ ☺

☰ ☷ ☹ ☺ 📄 🔗 📄 👁 📄 <>

Admin

p » span Mots : 1

Message affiché aux utilisateurs après avoir posté un commentaire :

Fichier ▾ Editer ▾ Insérer ▾ Voir ▾ Format ▾ Tableau ▾ Outils ▾

↶ ↷ 📄 Formats ▾ **B** *I* A ▾ A ▾ ☰ ☷ ☹ ☺

☰ ☷ ☹ ☺ 📄 🔗 📄 👁 📄 <>

Votre commentaire a bien été créé.
Il sera disponible une fois qu'il aura été validé pas un administrateur.

p Mots : 18

Modifier

Annuler

Votre opinion

Les options disponibles sont ici le choix d'une notification et la gestion du message d'information.

Configuration par défaut de l'extension

Votre opinion

Modifier la configuration par défaut

Message à afficher :

Fichier ▾ Editer ▾ Insérer ▾ Voir ▾ Format ▾ Tableau ▾ Outils ▾



Vous avez une remarque à faire, une expérience à raconter sur le contenu de la page ou sur le service dont il est question ?

p

Mots : 24

Notification par email * :

admin

Sélectionner la liste de diffusion à notifier.



Modifier



Annuler

GÉRER LES RÉSEAUX SOCIAUX

Si une liste des réseaux les plus courants est initialisée, vous pouvez en ajouter, en supprimer ou modifier les codes par défaut.

Gestion des réseaux sociaux

Ajouter un réseau social

Nb par page 50

Nom ^ ▾	Action
Facebook	 
Google+	 
Twitter (share)	 
Twitter (follow)	 

Comment configurer un réseau social ?

L'écran de modification, identique à l'écran de création, permet de définir le nom et le code javascript associé. Les trois zones, header, body et footer correspondent aux bookmarks que l'on retrouve dans le [modèle de page](#) comme page_frameset.html (head,body et footer).

Modifier un réseau social

Nom * Facebook

Contenu du header :

Contenu du body *

```
<!-- facebook -->
<div id="fb-root"></div>
<script>(function(d, s, id) {
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) return;
  js = d.createElement(s); js.id = id;
  js.src = "//connect.facebook.net/fr_FR/all.js#xfbml=1";
  fjs.parentNode.insertBefore(js, fjs);
})(document, 'script', 'facebook-jssdk');</script>
<div class="fb-like" data-href="http://dev.lutece.paris.fr/site-demo/" data-send="true"
data-layout="button_count" data-show-faces="true"></div>
```

Contenu du pied de page :

✓ Modifier ✕ Retour

Les réseaux sociaux choisis apparaitront sur le site à l'endroit que vous aurez déterminé lors de l'élaboration de votre charte graphique.

GÉRER LES TYPES DE VOTES

La liste des types de votes disponibles est initialisée par défaut mais vous pouvez la modifier avec vos propres paramètres.

Gestion des types de vote

Aide

Gestion des types de votes

ID	Titre	Actions
1	Star	
2	Thumb	
3	Vote simple	

Comment configurer un type de vote ?

Le contenu du modèle peut être complètement modifié, vous avez une aide dans la partie droite de l'écran qui vous donne la liste de bookmarks freemarker pouvant être utilisés dans votre code de modèle -contenu de la template-.

Modifier un type de vote

Titre *	<input type="text" value="Vote simple"/>														
Contenu de la template *	<pre><#if !voteClosed> <#if canDeleteVote> <div> J'annule mon vote </div> </#if> <#if show == "all" show == "actionVote"> <#if canVote> <div> <div class="extend-rating-vote-title"> <label for="">Votez :</label> </div> Je vote pour </div> </#if> </#if> </#if></pre>														
Labels freemarkers	<table><thead><tr><th>Nom</th><th>Freemarker</th></tr></thead><tbody><tr><td>ID de la ressource</td><td>`\${idExtendableResource!}`</td></tr><tr><td>Type de la ressource</td><td>`\${extendableResourceType!}`</td></tr><tr><td>Note totale</td><td>`\${rating.scoreValue!}`</td></tr><tr><td>Note moyenne (min : 0 - max : 4)</td><td>`\${rating.score!}`</td></tr><tr><td>Nombre de votes</td><td>`\${rating.voteCount!}`</td></tr><tr><td>Url pour voter</td><td><code>jsp/site/plugins/extend/modules/rating/DoVote.jsp?idExtendableResource=\${idExtendableResource!}&extendableResourceType=\${extendableResourceType!}&voteValue=<voteValue></code> avec <voteValue> à remplacer par une note positive ou négative.</td></tr></tbody></table>	Nom	Freemarker	ID de la ressource	`\${idExtendableResource!}`	Type de la ressource	`\${extendableResourceType!}`	Note totale	`\${rating.scoreValue!}`	Note moyenne (min : 0 - max : 4)	`\${rating.score!}`	Nombre de votes	`\${rating.voteCount!}`	Url pour voter	<code>jsp/site/plugins/extend/modules/rating/DoVote.jsp?idExtendableResource=\${idExtendableResource!}&extendableResourceType=\${extendableResourceType!}&voteValue=<voteValue></code> avec <voteValue> à remplacer par une note positive ou négative.
Nom	Freemarker														
ID de la ressource	`\${idExtendableResource!}`														
Type de la ressource	`\${extendableResourceType!}`														
Note totale	`\${rating.scoreValue!}`														
Note moyenne (min : 0 - max : 4)	`\${rating.score!}`														
Nombre de votes	`\${rating.voteCount!}`														
Url pour voter	<code>jsp/site/plugins/extend/modules/rating/DoVote.jsp?idExtendableResource=\${idExtendableResource!}&extendableResourceType=\${extendableResourceType!}&voteValue=<voteValue></code> avec <voteValue> à remplacer par une note positive ou négative.														
	<input type="button" value="Modifier"/> <input type="button" value="Annuler"/>														

GÉRER LES BOUTONS D'ACTIONS

La liste des boutons d'action disponibles est initialisée par défaut mais vous pouvez la personnaliser avec vos propres boutons.

Gestion des boutons d'actions

Gestion des boutons d'actions Aide

[+ Ajouter un bouton d'action](#)

Nombre par page

Nom <input type="button" value="v"/>	Ordre <input type="button" value="v"/>	Actions
Imprimer	1	<input type="button" value="p"/> <input type="button" value="v"/> <input type="button" value="d"/>
Envoyer	2	<input type="button" value="p"/> <input type="button" value="u"/> <input type="button" value="v"/> <input type="button" value="d"/>
Imprimer document	3	<input type="button" value="p"/> <input type="button" value="u"/> <input type="button" value="v"/> <input type="button" value="d"/>
Retour	4	<input type="button" value="p"/> <input type="button" value="u"/> <input type="button" value="d"/>

Comment configurer un bouton ?

Le contenu peut être entièrement personnalisé avec la possibilité de définir pour quel type de ressource (page, document, digg...) sera défini le bouton en utilisant les bookmarks correspondant à l'identifiant de la ressource -bookmark "@id_resource"- et/ou au type de ressource -bookmark "@type_resource"-.

Modifier un bouton d'action

Nom *

Type de ressource *

Le bouton d'action ne sera affiché que pour les extensions des ressources du type sélectionné.

Contenu

```
<a class="btn btn-small" target="_print" title="imprimer" href="jsp/site/plugins
/document/PrintDocument.jsp?document_id=@id_resource">|
<i class="icon-print"></i>
imprimer
</a>
```

Vous pouvez utiliser les paramètres suivants :

- @id_resource : Identifiant de la ressource
- @type_resource : Type de ressource

LES EXTENSIONS DANS LES MODÈLES HTML

Dans la *page_frameset.html*, le modèle de page principal de **Lutece**, les bookmarks "extend" sont présents afin d'être prêts à fonctionner sans faire d'autre modification. En fonction de votre charte, vous pouvez modifier l'emplacement des bookmarks notamment dans la partie "content".

```
<!DOCTYPE html>
<html lang="fr">
<head>
<base href="{base_url}">
<title>${favourite} - ${page_name}</title>
...
<#if isExtendInstalled?? && isExtendInstalled >
  <#if page_id??>
    @Extender[${page_id},PAGE,opengraph,{header:true}]@
  </#if>
  @Extender[ExtendParameteredId,document,opengraph,{header:true}]@
  @Extender[ExtendParameteredId,DIRECTORY_RECORD,opengraph,{header:true}]@
</#if>
...
</head>
<body id="home">
<!-- header -->
${page_header}
<!-- end header -->
<!-- Page path -->
${page_path}
<#if page_id?? && isExtendInstalled?? && isExtendInstalled >
  <!-- extends -->
  @Extender[${page_id},PAGE,opengraph,{header:false,footer:false}]@
  @Extender[${page_id},PAGE,hit,{show:true}]@
  @Extender[${page_id},PAGE,rating,{show:"all"}]@
  @Extender[${page_id},PAGE,actionbar]@
</#if>
<!-- content -->
${page_content}
<#if page_id?? && isExtendInstalled?? && isExtendInstalled>
  @Extender[${page_id},PAGE,comment]@
  @Extender[${page_id},PAGE,feedback]@
</#if>
<!-- footer -->
${page_footer}
<#if page_id?? && isExtendInstalled?? && isExtendInstalled >
  @Extender[${page_id},PAGE,opengraph,{footer:true}]@
</#if>
</body>
</html>
```

UTILISEZ LES EXTENDS DES CONTENUS DANS LES MODÈLES HTML ET LES XSL

Pour afficher un contenu particulier en front office, il faut ajouter le marker suivant dans une template :

@Extender[<idExtendableResource>, <resourceType>, <extenderType>, <parameters>]@

où :

<idExtendableResource> correspond à l'ID de la ressource (ex : 1) <resourceType> correspond au type de ressource. Exemples (attention, le type est sensible à la casse) :

Type de ressource	Description
PAGE	Il s'agit des pages classiques d'un portail Lutèce
document	Il s'agit des documents du plugin-document

<extenderType> correspond au type de l'extension. Exemples (attention, le type est sensible à la casse) :

Type d'extension	Description
comment	Pour commenter les ressources
hit	Pour le nombre de vues sur la ressource
rating	Pour voter les ressources
feedback	Pour donner son avis sur les ressources

<parameters> correspond aux paramètres supplémentaires à ajouter pour les extensions (ex : {show:true})

Quelques exemples :

Marqueur	Description
@ Extender[1,document,comment]@	Affiche les commentaires pour le document ayant l'ID 1.
@ Extender[1,document,hit,{show:true}]@	Affiche le nombre de vues sur le document ayant l'ID 1.
@ Extender[1,document,hit,{show:false}]@	N'affiche pas le nombre de vues pour le document ayant l'ID 1, mais incrémente le compteur.
@ Extender[1,document,hit,{increment:true}]@	incrémente le compteur
@ Extender[1,document,hit,{increment:false}]@	n'incrémente pas le compteur
@ Extender[1,document,rating,{show:"vote"}]@	Affiche la note du document ayant l'ID 1@
@	Affiche les liens pour voter le document ayant l'ID 1
Extender[1,document,rating,{show:"voteAction"}]@	Affiche la note et les liens du document ayant l'ID 1
@ Extender[1,document,rating,{show:"all"}]@	Affiche la note et les liens du document ayant l'ID 1
@ Extender[2,PAGE,feedback]@	Affiche un formulaire pour donner son avis sur la page ayant l'ID 2

Ci dessous un exemple de xsl qui inclut les informations issues des extends comment, hit et vote pour une liste de documents.

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:param name="site-path" select="site-path" />
  <xsl:param name="target" select="target" />
  <xsl:param name="page-id" select="page-id" />
  <xsl:variable name="portlet-id" select="portlet/portlet-id" />

  <xsl:template match="portlet">
    <div class="portlet">
      <xsl:apply-templates select="document-list-portlet/document" />
    </div>
  </xsl:template>

  <xsl:template match="document">
    <xsl:if test="not(string(document-xml-content)='null')">
      <a href="{ $site-path }?document_id={document-id}&#38;portlet_id={ $portlet-id } { $target }">
        <xsl:for-each select="descendant::*">
          <xsl:value-of select="document-title" />
        </xsl:for-each>
      </a>
      <!-- Inclusion des information des extend: Vote, Commentaires et Hits -->
      <xsl:if test="document-rating">
        
        (<xsl:value-of select="document-number-rating" />)
      </xsl:if>
      <xsl:if test="document-number-comment">
        <xsl:value-of select="document-number-comment" /> commentaires
      </xsl:if>
    </xsl:if>
  </xsl:template>
</xsl:stylesheet>
```

```
<xsl:if test="document-number-hits">
  <xsl:value-of select="document-number-hits" /> vues
</xsl:if>
</xsl:if>
</xsl:template>
</xsl:stylesheet>
```

HABILLAGE DES EXTEND

Il est possible de modifier le rendu graphique des "extends" pour les habiller selon votre charte.

Les fichiers modèles des modules sont placés dans cette arborescence WEB-INF/templates/skin/plugins/extends/modules/ où chaque sous répertoire contiendra les modèles à savoir:

- actionbar : Barre de boutons
- comment : Commentaires
- feedback : Votre opinion
- hit: Nombre de hit
- opengraph : Réseaux sociaux
- rating: Votes.

17. GÉRER L'AUTHENTIFICATION DU FRONT OFFICE

Lutece permet de réaliser des sites authentifiés soit complètement, soit seulement sur certaines pages front-office. Lutece propose par défaut plusieurs modes d'authentification que vous pouvez mettre en oeuvre en choisissant les modules développés pour le plugin MyLutece. Ce plugin gère l'ensemble des attributs communs à tous les modules d'authentification (portlet, attributs utilisateurs, ...).

Les types d'authentification par défaut sont les suivants:

- Base de données (login/mot de passe) -module mylutece-database;
- OpenAm -module mylutece-openam;
- FranceConnect - module mylutece-franceconnect
- CAS -module mylutece-cas;
- OAuth -module mylutece-oauth;
- OpenID -module mylutece-openid;
- LDAP - module-mylutece-ldapdatabase.

L'authentification permet de gérer des rôles qui pourront être associés à des utilisateurs et des ressources. Ainsi par exemple les pages ayant le rôle A ne pourront être visibles que des utilisateurs ayant le rôle A.

Le module d'authentification front-office va permettre de valoriser les attributs d'un objet LuteceUser : ses noms, prénoms, mail, adresse, ... par les informations ramenées par le type d'authentification du module (de la base de données pour database, d'un annuaire ldap, d'une authentification extérieure, ...).

PARAMÉTRER UN SITE AVEC MYLUTECE

Pour utiliser l'authentification, il faut activer via *Système > Gestion des plugins* le plugin MyLutece et le module sélectionné pour permettre aux utilisateurs d'accéder en mode authentifié au site. Par défaut le site n'est pas totalement authentifié, pour activer le mode, il faut modifier le fichier *WEB-INF/conf/plugins/mylutece.properties* et changer la valeur de la propriété **mylutece.portal.authentication.required** de *false* à *true*.

```
# Only authenticated users can access to the portal  
mylutece.portal.authentication.required=true
```

En fonction du module choisi, vous pourrez gérer les utilisateur dans le site (module-mylutece-database) ou en dehors du site.

En back-office, le menu "Utilisateur" permet d'accéder à la gestion des utilisateurs Front-office (à différencier de la gestion des utilisateurs back-office à laquelle on peut accéder via le menu "Gestionnaire").

Il est alors possible de gérer les utilisateurs, et de configurer leurs rôles, groupes, ...

The screenshot shows the Lutece back-office navigation bar with the 'Utilisateurs' menu highlighted by a red circle and an arrow pointing to the 'Gestion des rôles Lutèce' page. The page title is 'Gestion des rôles Lutèce' and it features a table of roles with columns for 'Libellé', 'Description', 'Groupe de travail', and 'Actions'. There is also an '+ Ajouter' button in the top right corner of the table.

Libellé	Description	Groupe de travail	Actions
wiki_admin	Wiki Administrator (can remove content)		 
wiki_contributor	Wiki Contributor (can edit content)		 
wiki_private	Wiki User (can see private content)		 

FILTRE D'AUTHENTIFICATION

Le plugin MyLutece apporte la possibilité de gérer des filtres d'authentification, c'est à dire de décrire, via le back-office, l'ensemble des pages accessibles sans authentification. Le nom des pages accessibles peut être soit exact, soit "jokerisé".

✦ Gestion du filtre d'authentification - Paramètres avancés

✔ Activer le filtre d'authentification Front Office		
Code	mylutece.url.login.page	URL <input type="text" value="jsp/site/Portal.jsp?page=mylutece&action=login"/> ✔ Modifier 
Code	mylutece.url.doLogin	URL <input type="text" value="jsp/site/plugins/mylutece/DoMyLuteceLogin.jsp"/> ✔ Modifier 
Code	mylutece.url.doLogout	URL <input type="text" value="jsp/site/plugins/mylutece/DoMyLuteceLogout.jsp"/> ✔ Modifier 
Code	mylutece.url.createAccount.page	URL <input type="text" value="jsp/site/Portal.jsp?page=mylutece&action=createAccount"/> ✔ Modifier 
Code	mylutece.url.modifyAccount.page	URL <input type="text" value="jsp/site/Portal.jsp?page=mylutece&action=modifyAccount"/> ✔ Modifier 
Code	mylutece.url.lostPassword.page	URL <input type="text" value="jsp/site/Portal.jsp?page=mylutece&action=lostPassword"/> ✔ Modifier 
Code	mylutece.url.lostLogin.page	URL <input type="text" value="jsp/site/Portal.jsp?page=mylutece&action=lostLogin"/> ✔ Modifier 
Code	mylutece.url.doActionsAll	URL <input type="text" value="jsp/site/plugins/mylutece/Do*"/> ✔ Modifier 

Nous allons ici prendre pour exemple la gestion de l'authentification via le module de gestion de base de données

GÉRER LES UTILISATEURS EN BASE DE DONNÉES

La gestion des utilisateurs du site peut se faire pour le module mylutece-database par le menu *Utilisateurs > Gestion des utilisateurs du site (database)*.

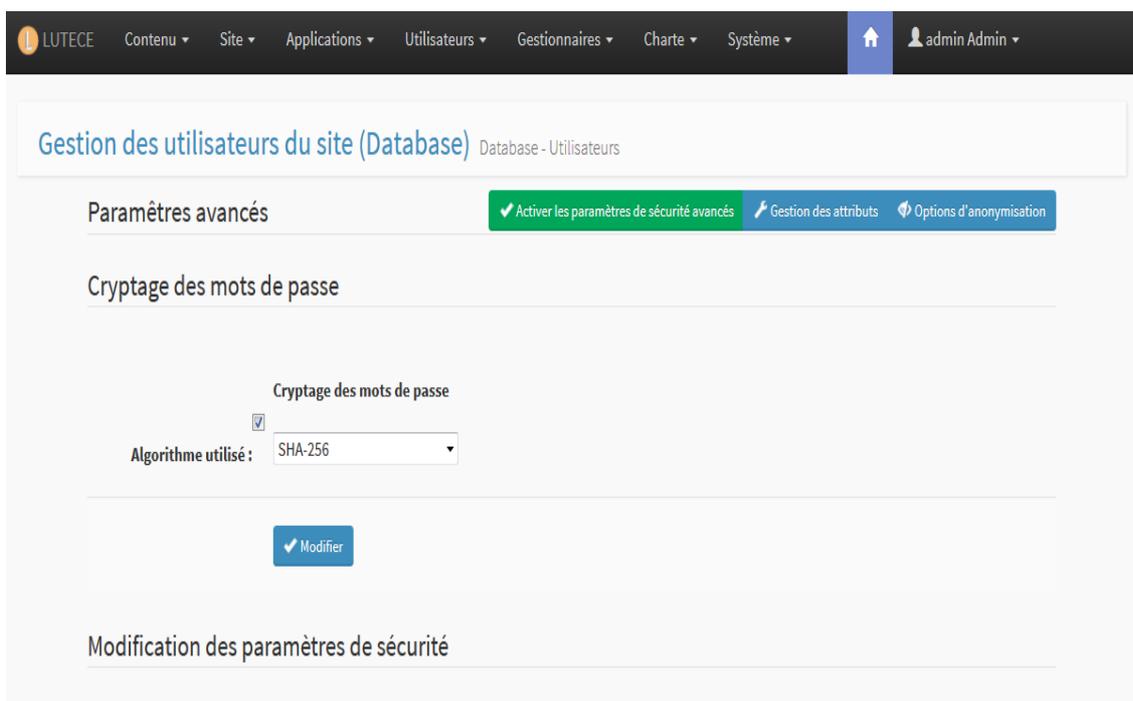
Il est possible d'effectuer les opérations suivantes:

- Gérer les comptes existants (modification des informations, suppression du compte, désactivation,)
- Créer un compte manuellement
- Gérer les rôles et groupes utilisateurs
- Importer ou exporter des utilisateurs
- Modifier les paramètres avancés (cryptage des mots de passe, connexion automatique, blocage des utilisateurs, ...)

LES PARAMÈTRES AVANCÉS

Les sites réalisés sur Lutece sont pour beaucoup d'entre eux des sites appartenant à une administration. Une partie des contraintes de sécurité et de protection des données liées aux contraintes réglementaires sont donc paramétrables dans les paramètres avancés du plugin mylutece.

Il est ainsi possible de sélectionner des paramètres tels que les algorithmes de cryptage des mots de passe, la validation d'une inscription par mail, de gérer une liste noire de domaines d'email, d'imposer une taille minimale de mot de passe, de tentatives de connexion, de durée de vie d'un compte utilisateur, ...



The screenshot shows the Lutece administration interface. At the top, there is a navigation menu with items like 'LUTECE', 'Contenu', 'Site', 'Applications', 'Utilisateurs', 'Gestionnaires', 'Charte', and 'Système'. The user is logged in as 'admin Admin'. The main content area is titled 'Gestion des utilisateurs du site (Database) Database - Utilisateurs'. Underneath, there is a 'Paramètres avancés' section with a green bar indicating 'Activer les paramètres de sécurité avancés'. Below this, there are tabs for 'Gestion des attributs' and 'Options d'anonymisation'. The 'Cryptage des mots de passe' section is expanded, showing a 'Cryptage des mots de passe' checkbox which is checked. Below it, there is a dropdown menu for 'Algorithme utilisé' currently set to 'SHA-256'. A 'Modifier' button is located at the bottom of this section. The next section is 'Modification des paramètres de sécurité'.

Validation par email

Cochez cette case si vous souhaitez que la création d'un compte soit activée par email

Connexion automatique après validation

Cochez cette case si vous souhaitez que l'utilisateur soit automatiquement connecté après validation son compte

Liste des noms de domaine interdits pour les email (séparés par des points virgules) :

yopmail.com

Taille minimale d'un mot de passe * :

8

Forcer le changement du mot de passe après une demande de réinitialisation

Nombre maximum de tentatives de connexion :

3

Durée du blocage des connexions :

10

Envoyer un email aux utilisateurs lorsqu'ils dépassent le nombre maximum de tentatives de connexion afin qu'il puissent se débloquer.

Durée de vie d'un compte utilisateur (en jours) :

360

Délai de renouvellement d'un compte (en jours) :

30

Relances avant expiration d'un compte :

2

Nombre de jours entre chaque relance :

10

 Modifier

Notifications

Modification d'une notification :

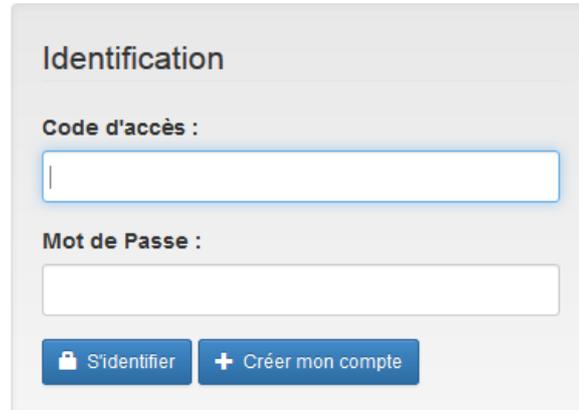
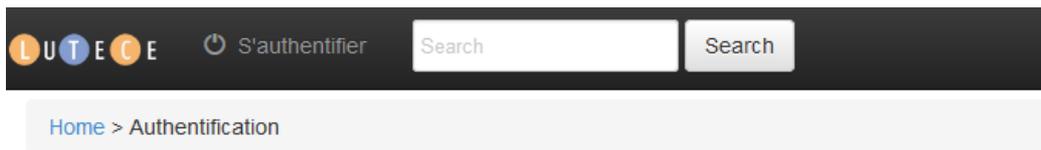
Première notification d'exp ▼

 Modifier l'email

Les contraintes d'authentification paramétrées dans le back-office s'appliquent pour la création de compte, et peuvent même entraîner une réinitialisation des mots de passe pour les comptes déjà créés dans le cas d'une modification de l'algorithme de cryptage du mot de passe.

ACCÉDER À UN SITE AVEC UNE AUTHENTIFICATION DEPUIS LE FRONT-OFFICE

Si le système d'authentification se fait par login/mot de passe, alors l'utilisateur qui cherche à s'authentifier, ou à accéder à un contenu authentifié, doit saisir ses informations dans la page de connexion.



The image shows a login form titled 'Identification'. It contains two input fields: 'Code d'accès :' and 'Mot de Passe :'. Below the input fields are two buttons: 'S'identifier' (with a lock icon) and '+ Créer mon compte'.

En front-office, l'utilisateur peut se voir proposer des écrans de gestion de son login oublié, du mot de passe oublié, la possibilité de créer son compte et de changer son mot de passe.

18. GÉRER LES UTILISATEURS BACK-OFFICE

Deux types d'utilisateurs sont à distinguer :

- Les **utilisateurs de l'interface d'administration Lutece** ou "**Gestionnaires Lutece**"
- Les **utilisateurs "visiteurs" du site public** ou "**Internautes**"

Dans ce chapitre nous ne parlerons que des "**Gestionnaires Lutece**". Ceux ci peuvent disposer de droits d'accès différents (les uns par rapport aux autres) aux fonctionnalités d'administration. La création et l'attribution de ces droits est **totale**ment paramétrable avec **Lutece**, ce qui permet de répondre aux spécificités de votre organisation interne.

Pour accéder à l'interface de gestion des utilisateurs, cliquer sur *Gestionnaires* > *Gestion des utilisateurs*. L'interface permet de :

- créer et gérer les utilisateurs
- leur assigner des droits
- leur assigner des rôles
- les regrouper dans des groupes de travail

Il est également possible d'importer et exporter des utilisateurs en masse depuis et vers des fichiers CSV. Nous détaillerons ci-dessous chacun de ces points.

La gestion des utilisateurs est composée de quatre grandes sections :

- La création des utilisateurs
- La modification des paramètres avancés de création et gestion des utilisateurs
- La recherche d'utilisateurs
- La liste des utilisateurs

CRÉATION D'UN UTILISATEUR

Le bouton d'action "Ajouter un utilisateur" amène à l'écran suivant :

The screenshot shows the 'Gestion des utilisateurs' interface with the 'Creation d'un utilisateur' form. The form is organized into three main sections: 'Identité', 'Compte', and 'Attributs complémentaires'.
- **Identité:** Includes fields for 'Nom *', 'Prénom *', and 'Email *'.
- **Compte:** Includes fields for 'Code d'accès *', 'Mot de passe *', and 'Confirmez le mot de passe *'. It also features a password complexity indicator showing '0%' and a 'Notifier l'utilisateur' section with radio buttons for 'Oui' and 'Non'.
- **Attributs complémentaires:** Includes dropdown menus for 'Statut' (set to 'Activé'), 'Niveau de droits' (set to 'Niveau 0 - Droits de l'administrateur technique'), 'Groupe de travail *' (set to '<Non assigné à un groupe>'), 'Langue' (set to 'français'), and 'Thème' (set to 'Mode accessible').
At the bottom center, there is a blue button labeled 'Ajouter un utilisateur'.

Cet écran gère l'identité du compte avec les trois champs :

- Nom
- Prénom
- Email

Les accès du compte avec :

- code d'accès (login)
- mot de passe
- confirmation du mot de passe
- un compte de complexité du mot de passe (le plus haut, le plus sécurisé)
- L'option de notifier l'utilisateur de la création de son compte (ou non)

Les attributs complémentaires :

- Statut (activé et valide ou désactivé et invalide du compte utilisateur)

- Niveau de droit : l'étendue des droits (de 0 le plus étendu, à 3 le plus restreint) : attention, cette option n'est pas modifiable une fois le compte créé. Cet indice détermine le niveau maximum des droits de l'utilisateur (par exemple s'il a un niveau de droit de 2, on ne pourra lui affecter que des droits de niveau 2 ou 3 sur les différents écrans et modules)
- Groupe de travail : l'appartenance ou non à un ou plusieurs groupes de travail. Les groupes de travail sont essentiellement utilisés pour restreindre l'accès de certaines ressources à un groupe déterminé d'utilisateurs, pour faire des mailing lists et communiquer par e-mail avec l'ensemble des membres d'un groupe.
- Langue : choix de la langue d'utilisation de Lutece (attention pour le moment il n'est possible de choisir que le français et l'anglais). Cette option ne fonctionne que si les ressources d'internationalisation ont été correctement paramétrées, dans le cas inverse, c'est le langage par défaut qui sera affiché.

Une fois l'ensemble des éléments complétés ou sélectionnés, valider la création du compte par le bouton "Ajouter un utilisateur" en bas de page.

IMPORT/EXPORT DES UTILISATEURS

L'export de la liste des utilisateurs vers un fichier CSV ou XML comprend la possibilité d'inclure ou non dans l'export :

- les attributs complémentaires des utilisateurs
- les rôles des utilisateurs
- les droits des utilisateurs
- les groupes de travail des utilisateurs

Gestion des utilisateurs Exporter des utilisateurs

Exporter des utilisateurs
Importer des utilisateurs

Feuille de transformation XSL à utiliser

Coeur - Export CSV des administrateurs
▼

Exporter les attributs complémentaires des utilisateurs
 Cocher cette case pour inclure les attributs complémentaires des utilisateurs dans le fichier d'export.

Exporter les rôles des utilisateurs
 Cocher cette case pour inclure les rôles des utilisateurs dans le fichier d'export.

Exporter les droits des utilisateurs
 Cocher cette case pour inclure les droits des utilisateurs dans le fichier d'export.

Exporter les groupes de travail des utilisateurs
 Cocher cette case pour inclure les groupes de travail des utilisateurs dans le fichier d'export.

✓ Valider
⏪ Retour

La page d'import d'une liste d'utilisateurs existants donne le détail de la structure requise pour le fichier CSV qui sera importé, propose d'ignorer la première ligne si celle-ci contient les titres des colonnes, et de mettre à jour les utilisateurs existants par l'import du fichier CSV.

Gestion des utilisateurs Importer des utilisateurs

Exporter des utilisateurs

Importer des utilisateurs

Fichier CSV * :

 Parcourir

Le séparateur de colonne du fichier CSV doit être ";".

Le fichier CSV doit contenir les colonnes suivantes : code d'accès, nom, prénom, email, statut (nombre entier), langue, niveau de l'utilisateur (nombre entier), obligation de changement de mot de passe (true ou false), mode accessible (true ou false), date de fin de validité du mot de passe, date de fin de validité du compte, date de dernière connexion puis les rôles, droits et groupes de travail, et enfin les attributs complémentaires. Utiliser la fonction d'export des utilisateurs au format CSV afin de visualiser un exemple.

Les attributs complémentaires sont facultatifs. Ils doivent être composés de l'identifiant de l'attribut, d'un caractère de séparation, de l'identifiant du champ, du caractère de séparation puis de la valeur de l'attribut. Le caractère de séparation doit être ":".

Les rôles, droits et groupes de travail doivent être composés d'une clé, du caractère de séparation des attributs complémentaires et de l'identifiant de l'objet (rôle, droit ou groupe de travail). La clé doit être 'role' pour les rôles, 'right' pour les droits et 'workgroup' pour les groupes de travail.

Attention, le fichier doit être encodé en UTF-8.

Ignorer la première ligne

Cocher cette case si la première ligne du fichier CSV contient les titres des colonnes et ne doit pas être importée

Mettre à jour les utilisateurs

Cocher cette case pour mettre à jour les utilisateurs existants.

Valider

Retour

GESTION DES PARAMÈTRES AVANCÉS

Les paramètres avancés offrent une sélection d'options supplémentaires pour gérer les comptes utilisateurs.

Activer les paramètres de sécurité avancés

Cryptage des mots de passe

Cryptage des mots de passe : Désactivé Activé

Algorithme utilisé : <Aucun algorithme>

Modifier

Modification des valeurs par défaut lors de la création d'un utilisateur

Niveau de droits : Niveau 0 - Droits de l'administrateur technique

Notifier l'utilisateur : Oui Non

Langue : français

Statut : Activé

Modifier

Modification du modèle de l'email

Configuration manuelle : Ne cochez pas cette case si vous souhaitez ajouter des règles de vérification.

L'écran est composé des éléments suivants :

- Le cadre de cryptage des données :
 - L'option d'activation ou de désactivation du cryptage
 - Le choix de l'algorithme de cryptage
- Le cadre des valeurs par défaut d'un utilisateur
 - Le niveau de droit par défaut à sélectionner dans la liste des droits possibles
 - La notification automatique de l'utilisateur
 - La langue par défaut
 - Le statut par défaut (activé ou non)
- Le cadre du modèle de l'email
 - Pour ajouter ou non des règles de vérifications
 - les règles de vérifications
 - Un bouton pour réinitialiser les contrôles
- Le cadre de modification des paramètres de sécurité
 - Un cadre pour gérer les noms de domaine d'e-mail interdits
 - La taille minimale d'un mot de passe
 - Le choix de forcer le changement de mot de passe ou non après une réinitialisation
 - Le nombre maximum de tentatives de connexions
 - La durée du blocage
 - La durée de vie d'un compte, au bout du délai le compte est désactivé
 - Le délai de renouvellement du compte (délai avant la fin du compte déclenchant une relance e-mail)
 - Le nombre de relance e-mail avertissant de l'expiration prochaine du compte
 - Le nombre de jours entre chaque relance
- Un cadre notifications qui permet de paramétrer le mail d'avertissement de l'expiration prochaine du compte
- 2 boutons pour accéder à deux autres menus (en haut à droite) :
 - Option d'anonymisation qui permet de sélectionner les champs à anonymiser par défaut parmi : le code d'accès (login), le nom, le prénom, l'e-mail. Mais aussi sur les champs ou attributs ajoutés par la gestion des attributs (ici par exemple le service d'appartenance)

Selection des champs à anonymiser

Identité

- Code d'accès
- Nom
- Prénom
- Email

Attributs complémentaires

Service d'appartenance :

Enregistrer

- La création de nouveaux attributs dans le formulaire de création de compte

Gestion des utilisateurs Gestion des attributs

Gestion des attributs Ajouter un nouvel attribut Texte

Titre	Type	Composant d'origine	Actions
Service d'appartenance	Texte	Coeur	 

- Pour ajouter un nouvel attribut, il faut choisir son type (texte, liste déroulante bouton radio, case à cocher, image) puis cliquer sur le bouton Ajouter. Cela amène à un autre écran.

Créer un champ de type liste déroulante

Titre *

Titre du champ

Aide(Formulaire de saisie)

Message d'aide pour l'interface des utilisateurs

- Présent dans la recherche
- Présent dans la liste des résultats
- Champ obligatoire
- Multiple
Sélection multiple de la liste

-
- Suivant le type choisi de formulaire, les éléments affichés seront légèrement différents. Il est nécessaire de renseigner les champs demandés puis de cocher les options souhaitées parmi celles proposées.
- Le nouvel attribut s'affichera dans la liste sur le menu précédent. Une fois dans la liste il est possible, à travers les boutons, de le modifier ou de le supprimer.

LA RECHERCHE D'UN UTILISATEUR

Gestion des utilisateurs Liste des utilisateurs

Liste des utilisateurs [+ Ajouter un utilisateur](#) [Importer/Exporter des utilisateurs](#) [Paramètres avancés](#)

Niveau de droits

Code d'accès Nom

Prénom Email

Statut

[Rechercher](#)

Ce cadre permet de filtrer et rechercher un utilisateur parmi la liste des comptes utilisateur existants. Il est possible d'effectuer une recherche et/ou de filtrer sur les critères suivants :

- Niveaux de droits : pour choisir l'ensemble des utilisateurs partageant le même niveau de droit
- Code d'accès : rechercher un login particulier
- Rechercher par le nom et/ou le prénom
- Rechercher par l'e-mail associé au compte
- Filtrer par statut

LA LISTE DES UTILISATEURS

Cet ensemble affiche la liste des utilisateurs par ordre alphabétique, par défaut les 50 premiers, ou le résultat d'une recherche de comptes.

Gestion des utilisateurs Liste des utilisateurs

Liste des utilisateurs [+ Ajouter un utilisateur](#) [Importer/Exporter des utilisateurs](#) [Paramètres avancés](#)

Niveau de droits

Code d'accès Nom

Prénom Email

Statut

[Rechercher](#)

Nom [^] _v	Prénom [^] _v	Code d'accès [^] _v	Email [^] _v	Statut	Actions
Admin	admin	admin	admin@exemple.fr	Activé	Modifier Verrouiller Réinitialiser Supprimer
...	Activé	Modifier Verrouiller Réinitialiser Supprimer
...	Activé	Modifier Verrouiller Réinitialiser Supprimer

La liste affiche :

- le nom de l'utilisateur
- le prénom
- le code d'accès
- l'e-mail du compte
- le statut activé ou désactivé
- Une série de boutons d'actions :
 - modifier le compte utilisateur
 - mettre à jour les droits
 - mettre à jour les rôles
 - gérer les groupes de travail

- anonymiser
- supprimer le compte

GÉRER LES DROITS DES UTILISATEURS

Après avoir créé le compte, il est nécessaire de lui assigner des rôles, des droits, et éventuellement de l'intégrer dans un groupe de travail. En cliquant sur le deuxième bouton "Droits", on accède à l'écran de paramétrage des droits. Pour le compte sélectionné, il est nécessaire de cocher la ou les cases de droits permettant d'accéder aux différents plugins.

Par exemple ici, la première case est cochée alors le compte pourra modifier la page d'accueil du site. C'est un droit de niveau 2 et donc le compte doit disposer au moins d'un niveau de droit de 2 (ou inférieur).

The screenshot shows the 'Gestion des utilisateurs' interface with the sub-header 'Modification des droits d'un utilisateur'. The user 'Admin admin' is selected. The 'Droits' tab is active, showing a list of permissions:

- Administration du Site (Niveau 2) - Création et mise à jour dynamique des pages, des rubriques et du Portail
- Administration générale (Niveau 2) - Administration générale du site
- Administration MyLutece (Niveau 2) - Administration MyLutece
- Agencement des menus du Back Office (Niveau 0) - Agencement des menus du Back Office
- Agencement des tableaux de bord fonctionnels (Niveau 0) - Gérer la mise en page des tableaux de bord d'administration fonctionnelle
- Agencement des tableaux de bord techniques (Niveau 0) - Gérer la mise en page des tableaux de bord d'administration technique

Valider avec le bouton en bas de la page pour confirmer les choix et enregistrer les nouveaux droits du compte.

GÉRER LES RÔLES D'UN UTILISATEUR

Les rôles sont des permissions qui donnent accès à la visibilité des boutons d'action.

De la même façon que pour les droits, il est nécessaire ici de cocher les cases des rôles voulus pour le compte puis de confirmer la sélection en validant.

Le détail de chaque rôle est visible dans *Gestionnaires > Gestion des rôles*.

Gestion des utilisateurs Modification des rôles d'un utilisateur

 Admin admin

Suivant →

Utilisateur Droits **Rôles** Groupes de travail

- all_site_manager - Site Manager
- assign_groups - Assigner des groupes aux utilisateurs
- assign_roles - Assigner des rôles aux utilisateurs
- content_contributor - Producteur de contenu
- content_publisher - Gestionnaire de publication

GÉRER LES GROUPES D'UN UTILISATEUR

Comme pour les deux autres onglets, il s'agit de sélectionner un ou plusieurs groupes de travail auxquels associer le compte utilisateur.

Gestion des utilisateurs

Modification des groupes de travail d'un utilisateur

 Admin admin

Suivant →

Utilisateur Droits Rôles **Groupes de travail**

Test Workgroup

✓ Valider

PROFILS D'UTILISATEURS

Pour les cas de sites ayant beaucoup d'utilisateurs back office, le **plugin Profiles** (plugin-profiles) permet de simplifier leur gestion en créant des profils. Ces profils regroupent des droits, des rôles et des groupes de travail prédéfinis, que l'on applique ensuite aux utilisateurs sans avoir à renseigner individuellement chaque compte. Il est également possible d'ajouter ou d'enlever manuellement certains droits/rôles/groupes de travail à un utilisateur sans toutefois lui enlever le profil.

19. GÉRER LES DROITS DES UTILISATEURS

NIVEAUX DE DROITS

Accessibles par *Gestionnaires > Gestion des niveaux de droit*, ces niveaux représentent les droits par profil utilisateur : administrateur technique, administrateur fonctionnel, webmestre et assistant webmestre. Le niveau 0 est celui qui possède le plus de droits, le niveau 3 celui qui en possède le moins.

Le choix parmi ces niveaux de droits est proposé au moment de la création d'un nouveau compte utilisateur.

Il est inutile d'ajouter de nouveaux niveaux de droits.

Gestion des niveaux de droit

Liste des niveaux de droits

Gestion des niveaux de droit + Ajouter

Identifiant	Libellé	Actions
0	Niveau 0 - Droits de l'administrateur technique	
1	Niveau 1 - Droits de l'administrateur fonctionnel	
2	Niveau 2 - Droits du webmestre	
3	Niveau 3 - Droits de l'assistant webmestre	

GESTION DES DROITS

Dans *Gestionnaires > Gestion des droits*, il est possible de voir à quel niveau (administrateur technique, fonctionnel, webmestre ou assistant webmestre) est associé un droit, et d'assigner des utilisateurs à un droit spécifique.

Gestion des droits

Liste des droits

Gestion des droits

Nom	Description	Niveau	Actions
Administration du Site	Création et mise à jour dynamique des pages, des rubriques et du Portail	2	
Administration générale	Administration générale du site	2	
Administration MyLutece	Administration MyLutece	2	
Agencement des menus du Back Office	Agencement des menus du Back Office	0	
Agencement des tableaux de bord fonctionnels	Gérer la mise en page des tableaux de bord d'administration fonctionnelle	0	
Agencement des tableaux de bord techniques	Gérer la mise en page des tableaux de bord d'administration technique	0	
Ajouter un fichier au serveur solR	Permet d'ajouter un fichier dans le serveur solR embarqué dans ce portail	2	

ASSIGNER UN UTILISATEUR À UN DROIT

Après avoir cliqué sur le bouton d'action "Assigner un utilisateur", la page suivante s'affiche :

Gestion des droits Assigner des utilisateurs à un droit

Administration du Site Niveau **2**  Nombre d'utilisateurs : **3**

Assigner des utilisateurs à un droit

← Précédent

Suivant →

Utilisateurs disponibles

bpico(BPICO BPICO)



Utilisateurs assignés

← Précédent

Suivant →

Nom ^ v	Prénom ^ v	Code d'accès ^ v	Email ^ v	Actions
Admin	admin	admin		 
				 
				 

Nombre par page 50  - Nombre d'éléments : 3

Les badges indiquent :

- le niveau du droit (ici 2)
- le nombre d'utilisateurs possédant ce droit (ici 2)

Les deux listes suivantes présentent :

- l'ensemble des utilisateurs disponibles à qui assigner ce droit
- les utilisateurs déjà assignés à ce droit

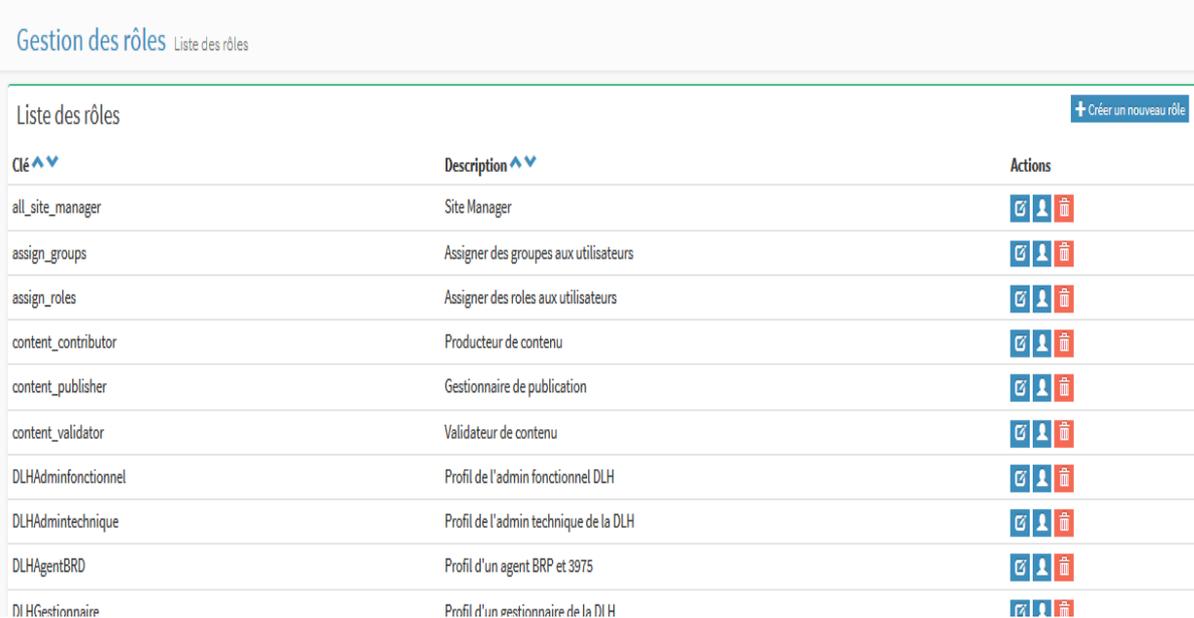
Pour assigner des utilisateurs au droit, il suffit de sélectionner le nom de l'utilisateur dans la liste des utilisateurs disponibles puis de valider pour lui conférer les permissions afférentes.

Dans la liste des utilisateurs assignés, il est possible d'éditer le compte de l'utilisateur avec le bouton , ou d'enlever le droit à un utilisateur avec le bouton .

20. GÉRER LES RÔLES DES UTILISATEURS

Pour accéder à la gestion des rôles, cliquer sur *Gestionnaires > Gestion des rôles*.

La page affiche la liste des rôles (qu'il est possible de trier en agissant sur les flèches bleues des en-têtes de colonnes) et propose la création de nouveaux rôles via le bouton dédié en haut à droite de l'écran.



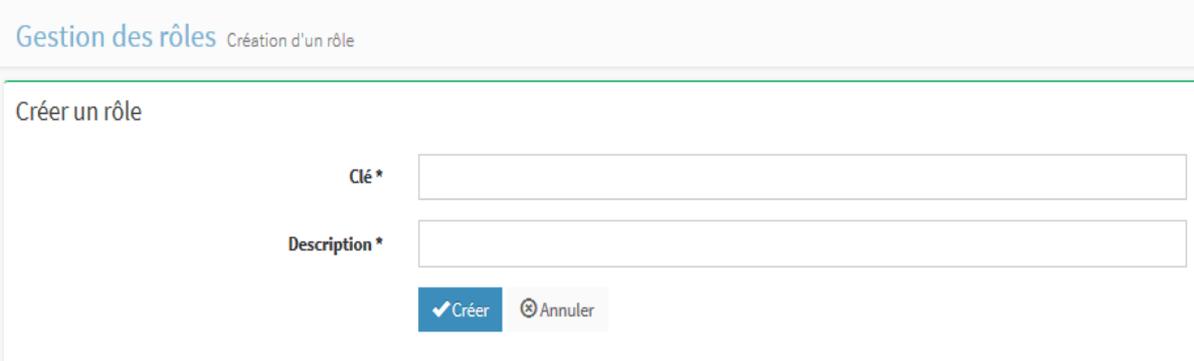
Gestion des rôles Liste des rôles

Liste des rôles + Créer un nouveau rôle

Clé	Description	Actions
all_site_manager	Site Manager	
assign_groups	Assigner des groupes aux utilisateurs	
assign_roles	Assigner des rôles aux utilisateurs	
content_contributor	Producteur de contenu	
content_publisher	Gestionnaire de publication	
content_validator	Validateur de contenu	
DLHAdminfonctionnel	Profil de l'admin fonctionnel DLH	
DLHAdmintechnique	Profil de l'admin technique de la DLH	
DLHAgentBRD	Profil d'un agent BRP et 3975	
DIHGestionnaire	Profil d'un gestionnaire de la DI H	

CRÉER UN NOUVEAU RÔLE

En cliquant sur le bouton "Créer un nouveau rôle", on affiche la page suivante :



Gestion des rôles Création d'un rôle

Créer un rôle

Clé *

Description *

Les deux champs à compléter sont :

- la clé : qui est le code de reconnaissance du rôle, ne doit pas contenir d'espaces, accents ou caractères spéciaux.
- la description : une description en toutes lettres du rôle

Puis on valide la création.

PARAMÉTRER LE NOUVEAU RÔLE

Une fois le nouveau rôle créé, s'affiche la liste de ses permissions, vide pour le moment, qui contient les champs suivants :

- A gauche le type de ressource
- Au milieu l'identifiant de la ressource : ***(*)** signifie l'ensemble des ressources
- A droite les permissions : ***(*)** signifie l'ensemble des permissions sur une ressource
- et tout à gauche le bouton de suppression du droit

il convient de le paramétrer afin de lui donner les permissions d'accès aux boutons d'action, au choix parmi une liste déroulante de composants, en haut à droite de l'écran.

Gestion des rôles Description d'un rôle

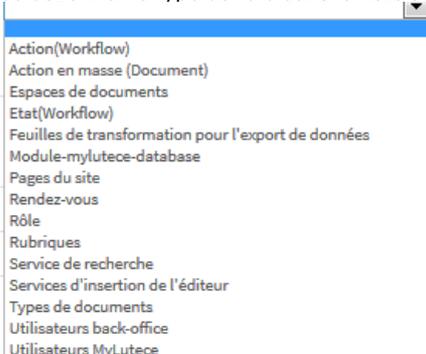
Admin_TEST - Rôles pour l'administrateur TEST

+ Modifier

Type de ressource	Identifiant de la ressource	Permission	Supprimer
-------------------	-----------------------------	------------	-----------

AJOUTER UNE NOUVELLE PERMISSION

- sélectionner le type de rôle dans la liste suivante (qui dépend des plugins activés lors de l'installation):



- Cliquer sur le bouton +

On accède ensuite au détail des permissions (dépendant du choix ci-dessus) :

- Il faut en premier sélectionner une ressource ou prendre toutes les ressources puis cliquer sur suivant :

Gestion des rôles Choix des contrôles

Ajouter un contrôle

- Sélectionner toutes les ressources de ce type
- Sélectionner les ressources parmi une liste

Suivant ➤

- Sélectionner une ou plusieurs ressources puis valider :

Gestion des rôles Sélection des permissions

Choisir les permissions

- Sélectionner toutes les permissions de ce type
- Sélectionner les permissions parmi une liste

- Modifier une fiche
- Copier une fiche
- Visualisation d'une fiche
- Créer un annuaire
- Modifier un annuaire
- Activer/Désactiver un annuaire
- Activer/Désactiver une fiche

21. GROUPES DE TRAVAIL

Il est possible de créer et gérer des groupes de travail via *Gestionnaires > Groupes de travail*. Le groupe de travail permet de restreindre l'accès à certaines ressources, comme un formulaire par exemple, à un groupe d'utilisateurs défini.

Groupes de travail

Liste des groupes

Liste des groupes

Nombre par page 50 - Nombre d'éléments : 21

Clé ^ v	Description ^ v	Actions
10E_ARRDT	10E_ARRDT	<input type="button" value="Modifier"/> <input type="button" value="Ajouter un utilisateur"/> <input type="button" value="Supprimer"/>
11E_ARRDT	11E_ARRDT	<input type="button" value="Modifier"/> <input type="button" value="Ajouter un utilisateur"/> <input type="button" value="Supprimer"/>
12E_ARRDT	12E_ARRDT	<input type="button" value="Modifier"/> <input type="button" value="Ajouter un utilisateur"/> <input type="button" value="Supprimer"/>
13E_ARRDT	13E_ARRDT	<input type="button" value="Modifier"/> <input type="button" value="Ajouter un utilisateur"/> <input type="button" value="Supprimer"/>

Le formulaire de création requiert une clé et une description :

Groupes de travail

Création d'un groupe de travail

Créer un groupe de travail

Clé *

Description *

De retour dans la liste des groupes de travail, vous pouvez assigner des utilisateurs au nouveau groupe créé via le bouton d'action .

Groupes de travail Assigner des utilisateurs à un groupe de travail

Assigner des utilisateurs à un groupe de travail **TEST**

 Nombre d'utilisateurs : **1**

← Précédent

Suivant →

Niveau de droits

<Tous les niveaux de droits>

Code d'accès

Nom

Prénom

Email

Statut

<Tous les statuts>

 Rechercher

Utilisateurs disponibles

← Précédent

Suivant →

Utilisateurs disponibles

✓ Valider

Utilisateurs assignés

← Précédent

Suivant →

Nom ^ v	Prénom ^ v	Code d'accès ^ v	Email ^ v	Actions
Admin	admin	admin		 

Pour ajouter un utilisateur au groupe de travail, le sélectionner dans la liste des utilisateurs disponibles et valider. Il apparaîtra ensuite dans la liste des utilisateurs assignés.

La liste des utilisateurs assignés propose le bouton d'action  pour modifier le compte de l'utilisateur ou  pour supprimer l'utilisateur du groupe de travail.

On peut ensuite les assigner à une ressource (page, formulaire, annuaire, document, accès à un espace de documents...). L'assignation se fait à partir de chaque ressource comme dans l'exemple ci-dessous, qui illustre les paramètres avancés d'un formulaire, via *Applications > Gestion des formulaires >  > Paramètres avancés*. L'onglet propose une liste déroulante "Groupe de Travail" qui référence les groupes disponibles.

Votants physiques campagne budget participatif 2015

Titre *

Titre court qui décrit le formulaire.

Description *

Descriptif plus complet du formulaire.

Enregistrer

Liste de questions

Paramètres avancés

Publication

Gestion des réponses

Front office: Titre

Permet de choisir le titre affiché pour ce formulaire dans le front-office.

Front office: Afficher le titre ?

Permet d'afficher ou non le titre ci-dessus en front-office.

Groupe de travail

Permet de restreindre l'administration du formulaire à un groupe de travail.

22. DÉFINIR UN WORKFLOW

QU'EST-CE QU'UN WORKFLOW DANS LUTECE ?

Le workflow désigne dans **Lutece** le circuit de traitement d'un dossier ou d'une demande. Une demande (ou un dossier) passe dans des **états** différents en fonction de l'avancement de son traitement et des **actions** effectuées.

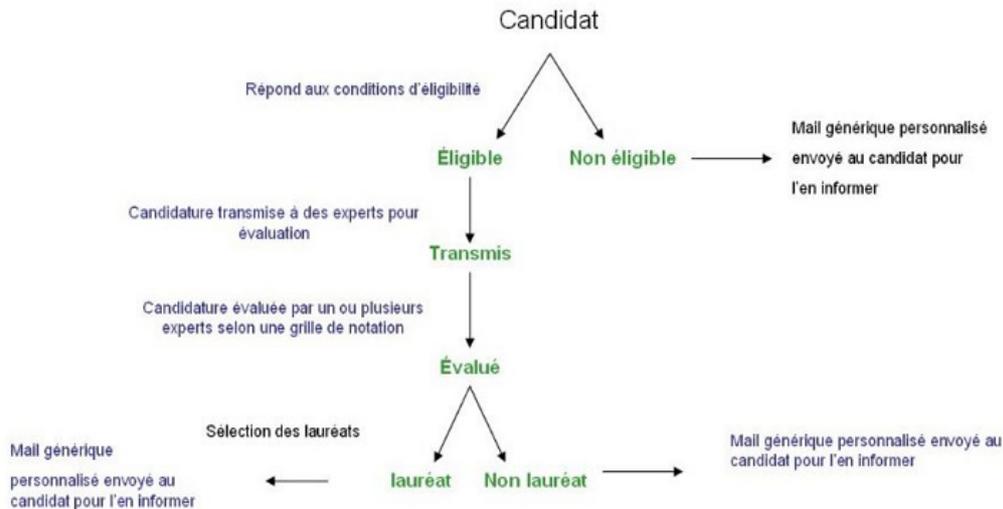
Avant de créer un workflow vous devez définir le circuit de traitement de votre dossier ou de votre demande.

Le plugin workflow est utilisé ici, pour la description des fonctionnalités, avec le plugin de gestion des données plugin-directory (annuaire).

Dans notre exemple, voir le schéma ci-dessous, les candidats postulent pour obtenir une bourse.

La collectivité a donc défini un circuit de traitement de la demande. Pour cela, l'administration vérifie si la candidature répond aux critères de recevabilité. Puis en fonction des critères de la demande, celle-ci est transférée au jury compétent pour évaluer les motivations du candidat.

Exemple de circuit de traitement fonctionnel de la demande



Dans le workflow Lutece, pour cet exemple les états seront les suivants :

- Nouvelle demande
- A traiter
- Éligible
- Non éligible
- Transmis
- Évalué
- Lauréat
- Non lauréat

Pour faire passer le dossier d'un état à l'autre, plusieurs actions vont être mises en oeuvre. Pour passer de l'état "*nouvelle demande*" à "*à traiter*", une action va être mise en place pour envoyer automatiquement un mail au demandeur avec un accusé réception de sa demande. De l'état "*nouvelle demande*", le dossier passera à "*traité*".

Une action manuelle permettra de passer de "*à traiter*" à "*éligible*" ou "*non éligible*", en fonction de la première étude qui est faite par la collectivité.

Les demandes sont dans le plugin utilisant le workflow (ici le plugin directory) sous forme de tableau et les boutons d'action présents au regard de chaque demande permettent d'agir sur cette dernière.

Exemple :

new test	Autre	Pas encore publiée	0	dsqd	Pétition à traiter	
nv	Autre	Pas encore publiée	0	Collectif	Pétition à traiter	

CONSTRUIRE UN WORKFLOW

Dans le menu de navigation, aller dans la "gestion des workflows". Cliquer sur créer un workflow.

Gestion des workflows

Statut : -Tous- Groupe d'utilisateur : «Non assigné à un groupe» Filtre

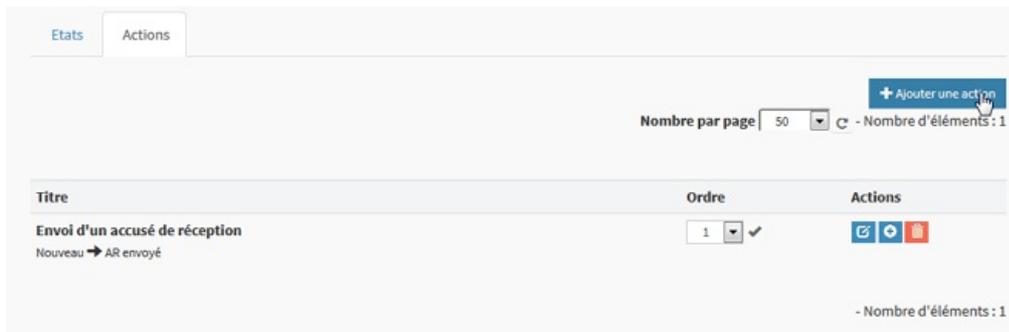
Paramètres avancés + Créer un workflow

Nombre par page : 50 Nombre d'éléments : 1

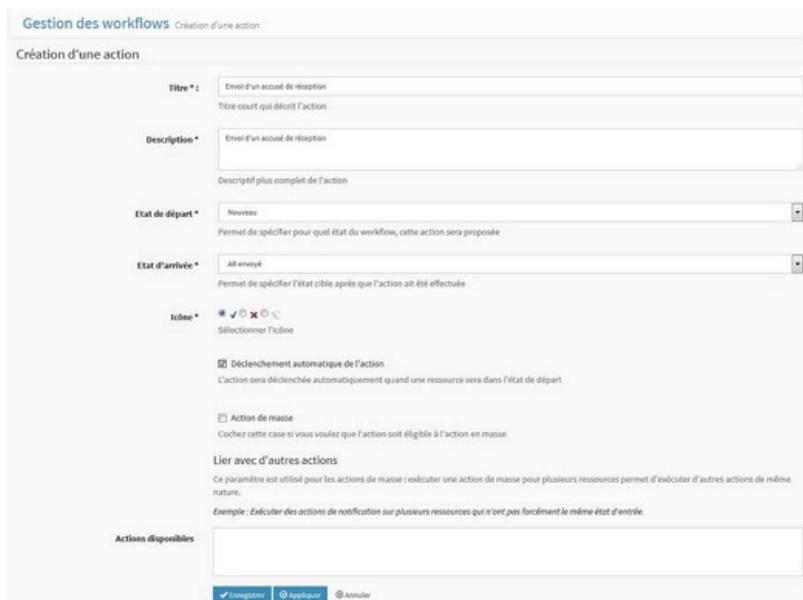
Titre	Actions
Valider	

- Nombre d'éléments : 1

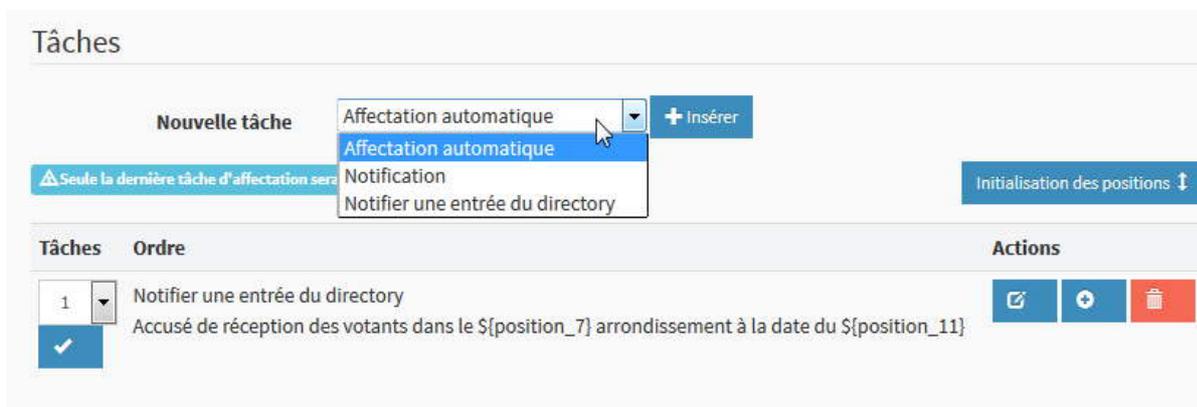
- d'un statut à l'autre
- Vous pourrez ensuite ajouter des tâches à cette action pour y ajouter des traitements automatiques
- Plusieurs types de tâches automatiques sont possibles. Il s'agit de modules supplémentaires à workflow qui doivent être assemblés si vous souhaitez vous en servir. Ces services ne sont pas fournis avec workflow seul.



Paramétrer l'action :



Modifier ensuite l'action pour y ajouter des tâches :



Plusieurs tâches peuvent être disponibles dans la liste :

- Affectation automatique permet par exemple d'attribuer une demande à un groupe de travail selon un critère déterminant. Par exemple, une demande d'un élève qui postule dans un cours d'art dramatique sera attribuée pour évaluation à un groupe de travail "jury d'art dramatique"
- Notifier une entrée du directory permet de configurer une tâche permettant d'envoyer un mail ou un sms si votre service le permet à la personne qui a déposé sa demande
- Notification permet de notifier un groupe de travail interne à l'application de l'arrivée d'une nouvelle demande par exemple.

Dans notre exemple nous allons utiliser "notifier une entrée du directory" pour envoyer un accusé de réception au demandeur

Créer et modifier une tâche

Sélectionner la tâche "notifier une entrée du directory" et modifier-la pour la configurer

De : Bureau de Paris (indivisible@paris.fr)
Envoyé : jeudi 6 août 2015 11:35
À : Bureau de Paris
Objet : Accusé de réception des votants dans le 2eme arrondissement à la date du 06/08/2015

Bonjour,

Nous avons bien reçu votre retour concernant l'urne n° Mobile.

Voici un récapitulatif des données que vous nous avez transférées :

N° : 1
Nom : [redacted]
Prénom : [redacted]
Email : [redacted]
Arrondissement : 2eme
Localisation de votre urne de vote :
Type d'urne : Mobile
Localisation de votre urne de vote :
Votes correspondants à la date du : 06/08/2015
Nombre de personnes ayant voté : 100
Nb de votants pour tout Paris : 50
Nb de votants pour les projets localisés : 50

Merci à vous,

UTILISER UN WORKFLOW POUR EXPORTER UN FORMULAIRE OU UNE ENQUÊTE DANS UN ANNUAIRE

La mise en place et l'utilisation d'un annuaire comporte de multiples avantages, ainsi vous pouvez :

- ajouter la date de création de chacun des enregistrements,
- ajouter des champs de recherche, modifier le libellé d'un champ ou sa présence,
- ajouter des champs pour traiter la demande,

Le workflow sera mis en place en suivant ces étapes :

1. Créer le formulaire
2. Gérer les processus externes dans le formulaire et créer l'annuaire
3. [Créer le workflow](#) et associer l'annuaire et le champ e-mail de l'annuaire pour envoyer un e-mail
4. Associer le workflow à l'annuaire dans les paramètres avancés de l'annuaire.

23. PARAMÉTRER UN PROCESSUS DE GESTION DE DEMANDE

Pour illustrer un processus de gestion de demande avec Lutèce, nous allons nous appuyer sur le projet de mise en place d'un guichet permettant de centraliser un ensemble de demandes et de types de demandes (qui peuvent être sur des sites distants ou bien sur le même site).

Exemple :

- un formulaire est un type de demande (plugin-form, plugin-appointment pour une demande de rendez-vous)
- une réponse au formulaire est une demande

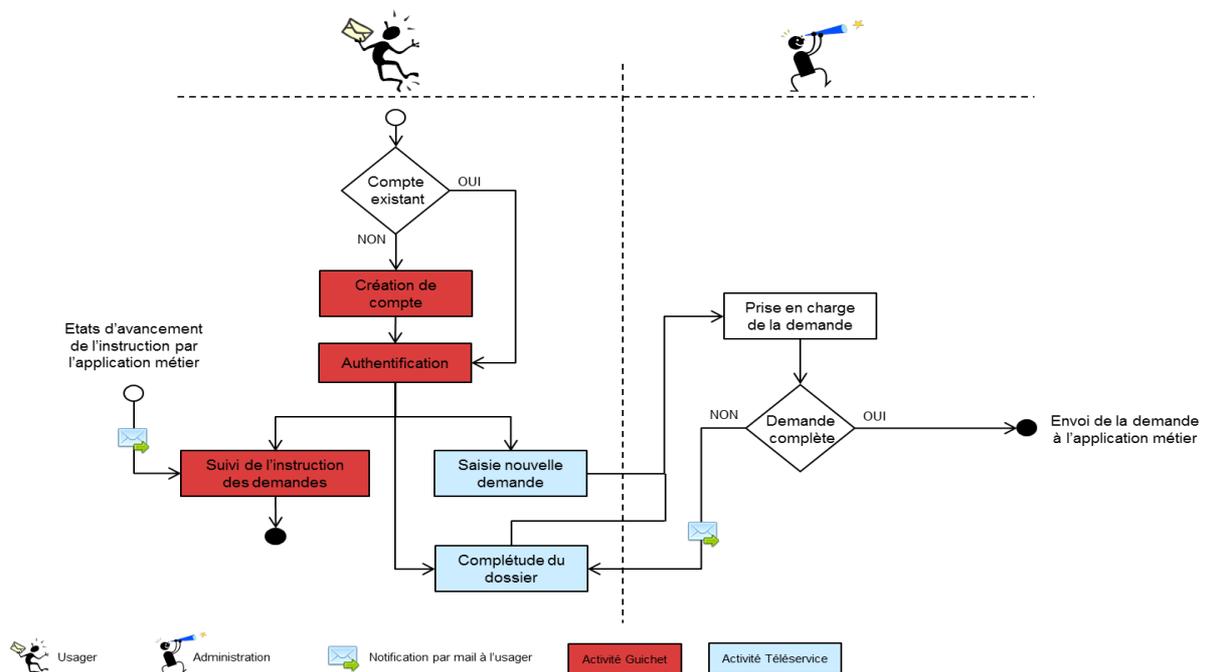
Ce cas d'utilisation répond au besoin où les gestionnaires doivent interagir avec les utilisateurs Front-Office sur leurs demandes.

Le Guichet est un simple portail permettant à l'utilisateur :

- d'accéder aux différentes démarches administratives en ligne de la ville de Paris via des téléservices,
- de remplir ses demandes en plusieurs fois en enregistrant éventuellement des brouillons,
- de retrouver l'historique de ses demandes dans un espace personnel,
- d'être informé de l'état d'avancement de l'instruction de ses demandes.

L'accès à la majorité de ces services est soumis à authentification au moyen d'un compte que l'utilisateur aura préalablement créé en ligne.

Le diagramme suivant résume l'exemple ci-dessus :



La partie front-office se présente sous cette forme à l'utilisateur :

MAIRIE DE PARIS

GUICHET DES PROFESSIONNELLS DE PARIS

[S'authentifier](#)

Accueil [Mes demandes](#) [Mes brouillons](#) [Mon compte](#)

Accéder à votre compte



Avec le Guichet des Professionnels de Paris vous pouvez :

- Accéder aux différentes démarches administratives en ligne de la ville de Paris
- Remplir votre demande en plusieurs fois en enregistrant des brouillons
- Retrouver l'historique de vos demandes dans votre espace personnel
- Etre informé de l'état d'avancement de l'instruction de vos demandes

Créer votre compte dès maintenant

[S'inscrire](#)

Autorisation d'occupation du domaine public

Stationnement

- Demande d'enseigne
- Demande de terrasse ou étalage
- Demande de cendrier mobile

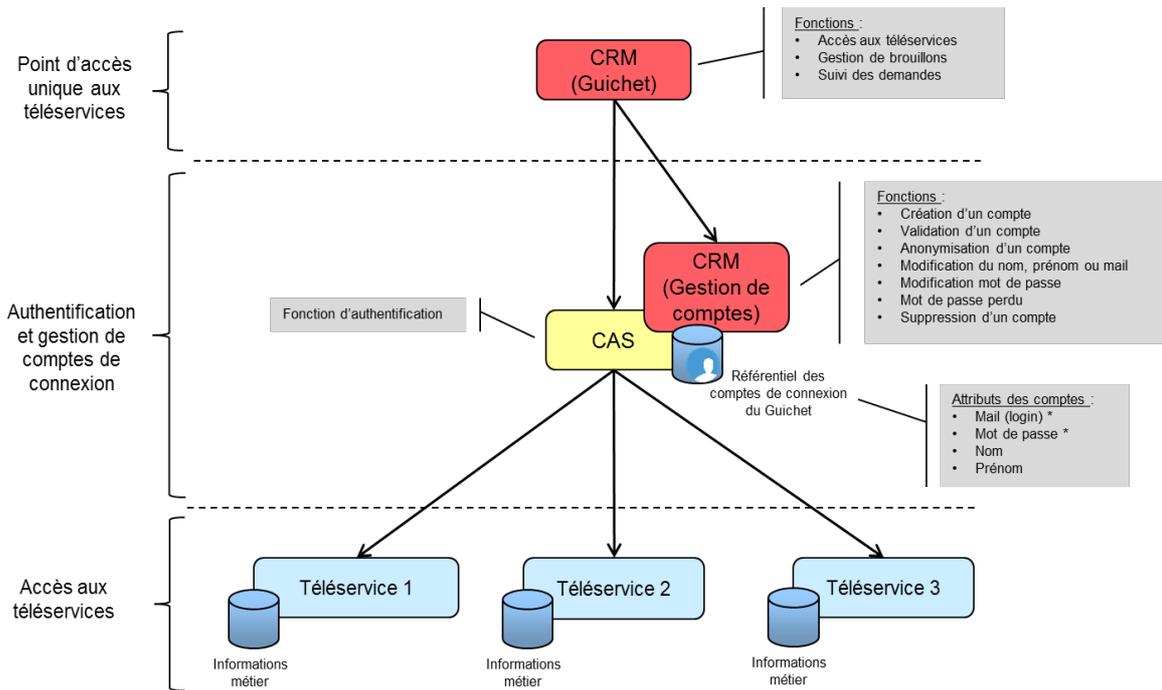
• Demande de cartes de stationnement



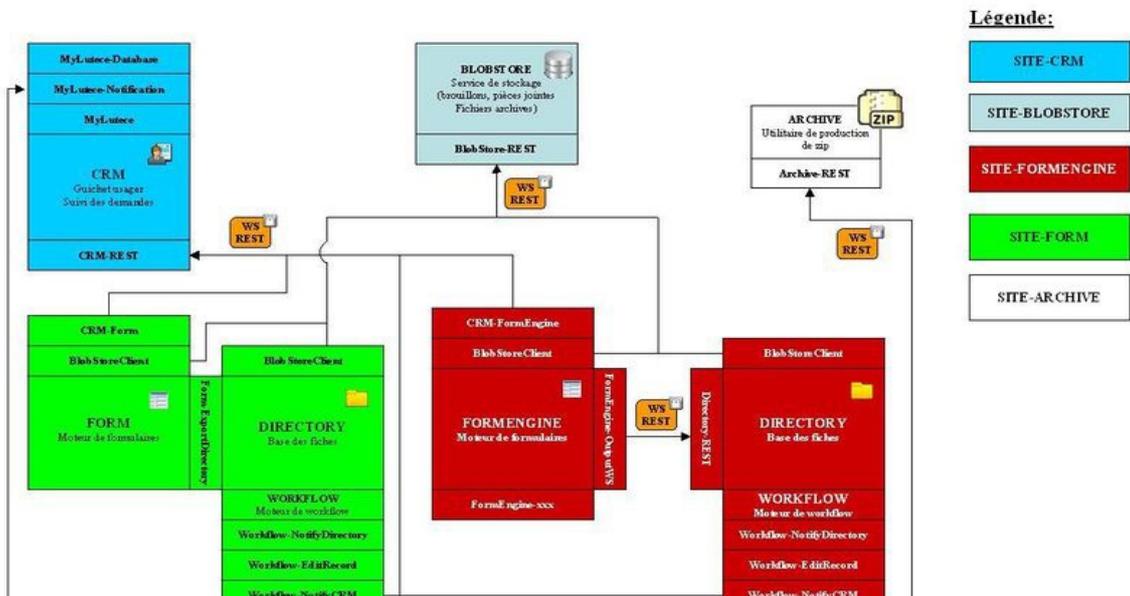
ARCHITECTURE

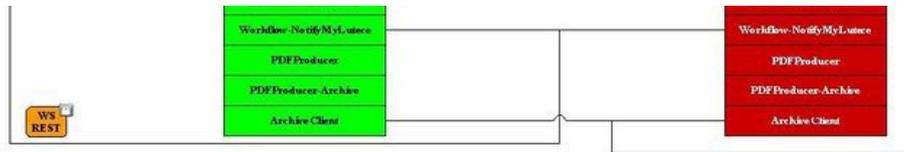
Il est possible de coupler ces composants avec tout type d'authentification, l'idéal étant d'utiliser un module d'authentification permettant à l'utilisateur de ne se connecter qu'une fois (SSO), pour pouvoir naviguer d'un téléservice à un autre.

Il est souvent plus intéressant de séparer les fonctionnalités entre plusieurs instances de Lutèce, pour faciliter les opérations de maintenance, l'évolution des composants, l'arrêt ou la mise en place d'un nouveau service n'ayant pas d'impact sur ceux déjà en place.



Dans l'exemple ci-dessous, le schéma présente la création de diverses instances qui ont été nécessaires à la réalisation d'un site de demandes avec suivi. Ce projet utilise à la fois des plugins standard et paramétrables comme les formulaires (plugin form) et des plugins développés pour des besoins particuliers. Il est tout à fait envisageable d'intégrer d'autres plugins pour répondre à d'autres besoins, comme par exemple le plugin appointment pour intégrer une demande de rendez-vous dans son portail d'accès.





ETUDE DE CAS : METTRE EN PLACE UNE PLATE-FORME D'APPELS À PROJETS

La présente étude de cas porte sur une plate-forme d'appels à projets réalisée dans le cadre du dispositif « Paris Jeunes ». Elle couvre à la fois la partie dépôt des dossiers en ligne par les candidats et l'instruction des dossiers par les gestionnaires et les différents jurys.

Interface candidat

Le candidat doit pouvoir accéder au dépôt et au suivi de son ou ses dossiers par le biais d'un compte unique authentifié, commun à l'ensemble des dispositifs et qu'il peut se créer lui-même. Il doit pouvoir créer un ou plusieurs dossiers de candidature, les visualiser dans leur état en cours ou validé, les compléter ou les supprimer.

La saisie des dossiers doit pouvoir se faire par étape, avec une sauvegarde d'un dossier à l'état brouillon à chaque étape. Ce brouillon doit être accessible pour être complété lors d'une connexion ultérieure.

Les dossiers pouvant comprendre des pièces jointes lourdes (documents audio ou vidéo), le téléchargement de ces documents doit pouvoir se faire au fil de l'eau et non au moment de la validation finale. De plus l'interface de chargement doit présenter une jauge de progression permettant à l'utilisateur de suivre la bonne réalisation du téléchargement.

La gestion du compte utilisateur doit prévoir : l'activation du compte pour retour d'email, gestion du mot de passe perdu par procédure de réinitialisation à partir d'une URL temporaire.

Interface de gestion

Les gestionnaires ont accès à l'ensemble des dossiers validés par les candidats. Ils peuvent demander des compléments d'informations aux candidats, rejeter des dossiers ne remplissant pas les conditions d'éligibilité. Ils peuvent télécharger pour un ou plusieurs candidats un dossier sous forme d'un fichier d'archive contenant un document PDF contenant l'intégralité des informations du candidat et l'ensemble des pièces annexes comme les œuvres à juger.

Interface Jury

Les membres du jury appartiennent à différentes commissions et disposent d'un compte d'accès pour visualiser tous les dossiers des candidats concourant dans leurs commissions. Ils peuvent télécharger pour chaque candidat un dossier sous forme d'un fichier d'archive contenant un document PDF contenant un extrait des informations du candidat et certaines pièces annexes comme les œuvres à juger.

SOLUTION MISE EN ŒUVRE

Interface candidat

L'interface candidat est architecturée autour des plugins suivants :

- MyLutece pour l'authentification
- CRM pour la gestion de la création des dossiers, de la gestion des dossiers en cours et du suivi des dossiers validés.
- FormEngine comme moteur de formulaire à étapes. Chaque formulaire correspondant à un des dispositifs est un module de Formengine. Il serait possible d'utiliser le plugin-form pour un cas de formulaire sans validation intermédiaire entre les étapes, par simple paramétrage.

Interface gestionnaires

Les dossiers validés sont transférés dans le plugin Directory qui permet leur suivi par un workflow et toutes les opérations d'édition et d'extraction. Le transfert est réalisé par des Web services REST mettant en œuvre le module OutputWS de FormEngine et le module REST de Directory.

Interface Jury

L'interface Jury est réalisée par un plugin spécifique qui interroge la base des dossiers gérée par Directory à l'aide de webservices REST et filtre les dossiers en fonction du rôle de l'utilisateur.

ÉLÉMENTS SPÉCIFIQUES

Les éléments spécifiques à la plate-forme sont les trois formulaires FormEngine et l'interface de consultation des dossiers du jury. La charge de réalisation de ces éléments a été de 8 jxh par formulaire et 12 jxh pour l'application de consultation.

ÉLÉMENTS GÉNÉRIQUES

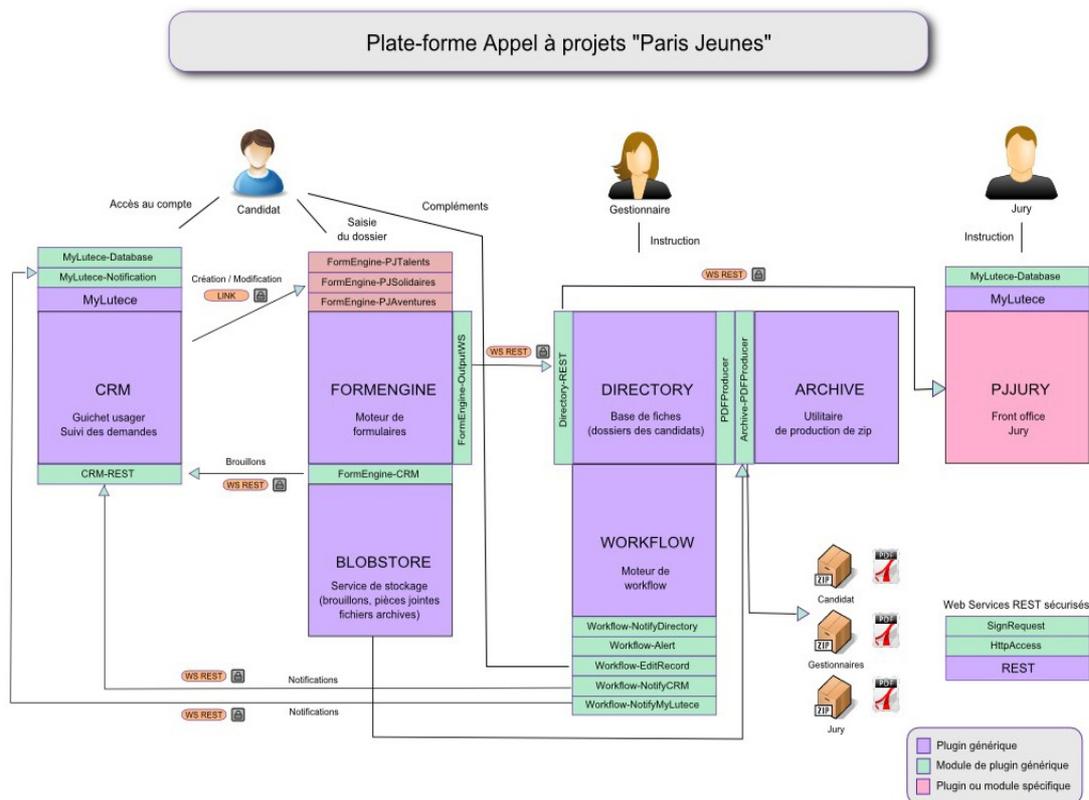
Liens documentaires de référence :

- <http://dev.lutece.paris.fr/plugins/plugin-formengine/fr>
- <http://dev.lutece.paris.fr/plugins/module-formengine-outputws/fr>
- <http://dev.lutece.paris.fr/plugins/plugin-directory/fr>
- <http://dev.lutece.paris.fr/plugins/plugin-workflow/fr>

Liens documentaires vers guides d'intégration :

- http://wiki.lutece.paris.fr/lutece/isp/site/Portal.jsp?page=wiki&page_name=howto_rest&view=page

SCHÉMA GÉNÉRAL D'INTÉGRATION



CONFIGURATION DU POM.XML

Pour créer un site de suivi de demandes, un assemblage de plugins est nécessaire, certains obligatoires pour les fonctionnalités de base, d'autre en option pour augmenter le nombre de fonctionnalités du site.

Composants obligatoires

Composant	Description
Core	Composant coeur
plugin-crm	"Client Relation Management" : Ce composant permet de réaliser la centralisation des types de ressources et des ressources
plugin-rest	Ce composant permet d'établir des communications avec les sites externes par la représentation architecturale REST. Voir le Guide de développement Lutece - chapitre Créer des Web Services Rest pour plus d'information.
module-crm-rest	Ce composant permet de recevoir des communications externes pour le plugin-crm, en l'occurrence pour une mise à jour des ressources, pour un envoi d'une notification d'une ressource.
plugin-mylutece	Ce composant permet d'avoir une gestion des utilisateurs en Front-Office du site Guichet Unique, ce qui est nécessaire pour regrouper un ensemble de ressources pour un utilisateur donné.
module-mylutece-xxx	Le plugin-mylutece ne peut fonctionner seul. Il faut également un module d'authentification MyLutece. Un exemple de module simple est le module-mylutece-database.
library-httpaccess	Cette bibliothèque permet de réaliser des opérations par Web Service. Le plugin-crm en est dépendant.
library-signrequest	Cette bibliothèque fournit une API pour définir des services d'authentification de requêtes HTTP. En d'autres termes, elle est utilisée pour avoir une sécurité sur les communications entre les diverses applications par Web Service REST. La library-httpaccess en est dépendante.

Composants optionnels

Composant	Description
module-mylutece-notification	Ce composant permet aux utilisateurs MyLutece d'avoir une boîte au lettre générique qui leur permet de recevoir des notifications venant de l'application ou bien venant d'autres utilisateurs MyLutece.
module-mylutece-rest	Ce composant permet aux applications externes de récupérer les informations d'un utilisateur (ex : nom, prénom, email, etc...).

N.B.: Vous pouvez bien entendu ajouter d'autres plugins, comme par exemple le plugin-html, pour la création de contenu.

Vous retrouverez un exemple de POM en [Annexe 3](#)

CONFIGURATION DES FICHIERS PROPERTIES ET DES FICHIERS XML DES PLUGINS

Pour configurer les divers plugins, il faut surcharger les fichiers de propriétés (avec seulement les propriétés à modifier) dans le dossier **WEB-INF/conf/override/plugins/**, et les fichiers XML des plugins dans les dossiers **WEB-INF/conf/plugins/** et **WEB-INF/plugins/**

Les fichiers des plugins sont à configurer pour qu'ils puissent entrer en interaction, notamment pour la communication des webservices.

Les fichiers `crm_context.xml`, `crm-rest.xml`, `crm.properties`, `httpaccess.properties`, `mylutece.properties`, `mylutece-rest.xml` sont également à configurer.

Exemple du fichier `crm_context.xml` (configuration de la clé privée) :

```
<!-- SignRequest -->
<bean id="crm.hashService" class="fr.paris.lutece.util.signrequest.security.Sha1HashService" />
<bean id="crm.requestAuthenticatorForWS"
class="fr.paris.lutece.util.signrequest.HeaderHashAuthenticator" >
  <property name="hashService" ref="crm.hashService" />
  <property name="signatureElements" >
    <list>
    </list>
  </property>
  <property name="privateKey">
    <value>change me</value>
  </property>
</bean>
<bean id="crm.requestAuthenticatorForUrl"
class="fr.paris.lutece.util.signrequest.RequestHashAuthenticator" >
  <property name="hashService" ref="crm.hashService" />
  <property name="signatureElements" >
    <list>
    </list>
  </property>
  <property name="privateKey">
    <value>change me</value>
  </property>
</bean>
```