
Moteur de rendu inJAC

Installation

Table des matières

Pré-requis	1
Installation	1
deployer l'application	1
Configuration générale	2
Configuration du filtre CAS	3
Configuration avancée	4
Niveau de débogage	4
Personnalisation	5
Personnalisation du rendu	5
Localisation	8

Pré-requis

Sur la machine de dépoïement ou sur d'autres machines :

Le moteur de rendu inJac est le rendu d'un espace de publication Webdav, il dépend donc d'Esup-Portail, du Canal de Stockage et du serveur Webdav.

En outre, l'accès authentifié nécessite l'installation d'un serveur CAS(CAS server with CAS GH ou CAS QuickStart). Tous sont disponibles sur l'espace de téléchargement , respectivement dans les sections 'Espace de Stockage' et 'SSO'.

Ceux-ci doivent être déployés, soit sur la même machine, soit sur des machines accessibles par réseau à la machine sur laquelle est déployée l'application.

Sur la machine de déploiement du moteur de rendu :

Sur la machine de déploiement doivent être installés un environnement Java (JRE ou JDK), version 1.4.2 ou supérieur, ainsi qu'un moteur de servlet/jsp (TOMCAT 5.0.28 ou supérieur).

Sous **Linux**, cette application nécessite d'installer les bibliothèques X, et lors du lancement de la jvm d'ajouter le paramètre :

```
-Djava.awt.headless=true
```

Sous Windows 98/2000/XP, aucune configuration supplémentaire n'est requise.

Note : L'application n'a pas été testée sous Windows XP Edition 64 bits.

Installation

deployer l'application

Pour déployer le moteur de rendu inJAC, il suffit de décompresser l'archive dans un moteur de servlet. Pour Tomcat, l'application peut être déployée dans le répertoire **/webapps**.

Configuration générale

Une fois déployée, l'application se configure en éditant le fichier `iconf.xml` dans le répertoire `injac/conf`.

- Les URL

```
<!-- Webdav (hostname+port+path) authenticated mode -->
<webdav>slide-priv.enseeiht.fr:8080/slide/files/injac/injacRoot</webdav>
<!-- URL Webdav (hostname+port+path) public mode -->
<anon-webdav>slide-pub.enseeiht.fr:8080/slide/files/injac/injacRoot</anon-webdav>
<casURL>https://cas.enseeiht.fr:8443/cas</casURL>
<!-- URL of Injac application (don't use 'localhost', but DNS or IP-->
<baseURL>http://saroumane.enseeiht.fr:810/injac</baseURL>
<!-- URL of Injac-indexing application (don't use 'localhost', but DNS or IP-->
<indexingURL>http://saroumane.enseeiht.fr:810/injac-indexing</indexingURL>
```

<webdav> et <anon-webdav> Avant toute chose, vous devez définir les url du serveur Webdav, ainsi que l'url de votre application Injac. Pour gérer les deux mode d'accès, public et avec identification, deux url distinctes sont nécessaires, avec deux *hostnames* différents, un pour chaque mode. Ces *hostnames* doivent correspondre à ceux définis dans les filtres "TRUSTED" et "ANON" du serveur Webdav (fichier `web.xml` de l'application Slide) :

```
<filter>
<filter-name>authenticationRouter</filter-name>
...
<!-- TRUSTED -->
<init-param>
<param-name>org.esupportail.filter.authenticationRouter.destinationHostTRUSTED</param-name>
<param-value>slide-priv.enseeiht.fr:8080</param-value>
</init-param>
<!-- UNAUTHENTICATED -->
<init-param>
<param-name>org.esupportail.filter.authenticationRouter.destinationHostUNAUTHENTICATED</param-name>
<param-value>slide-pub.enseeiht.fr:8080</param-value>
</init-param>
...
</filter>
```

Ces URLs doivent correspondre à celles définies dans `iconf.xml` (voir plus haut caractères gras).

<baseURL> correspond à l'URL à laquelle le moteur de rendu va être accessible.

<indexingURL> correspond à l'URL à laquelle le moteur de rendu va être accessible.

- Espace racine

<upnode> est le nom WEBDAV du répertoire racine de l'arborescence de navigation. Il correspond **impérativement** au dernier élément des URL WEBDAV définies plus haut. <display-upnode> est

le nom affiché pour ce répertoire racine.

- Paramètre d'affichage et d'accessibilité

<**enable-external-preview**> booléen pour autoriser ou non la prévisualisation de documents dont le rendu est externe(Word, Open Office, RTF, PDF...).

<**trustedpw**> mot de passe TRUSTED pour la connexion au serveur Webdav(doit être le même que celui défini dans le filtre TRUSTED du serveur WEBDAV.

<**locale**> langue utilisée pour la navigation. Les langues implémentées dans cette version sont le français(fr) et l'anglais(en). (Pour configurer le module de traduction, se reporter à la section Localisation.)

<**access mode**> mode d'accès au moteur de rendu :

- public : seul l'accès anonyme est utilisé. Pas de lien pour s'authentifier.
- private : seul l'accès authentifié est utilisé. Pas d'accès anonyme.
- mixed : les deux modes sont accessibles.
(public|private|mixed)

<**errorsheet**> Feuille utilisée pour signifier les erreurs. A changer uniquement pour le débogage.

- Paramètres liés aux métadonnées WEBDAV (**configuration avancée**)

Ces paramètres assurent l'interfaçage avec le Canal de stockage. Les valeurs par défaut correspondent à la version courante de ce dernier, contemporaine de la sortie de ce package. **Ne pas toucher sauf en pleine connaissance de cause !**

Configuration du filtre CAS

L'accès authentifié se fait par l'intermédiaire d'un filtre java, à l'aide d'un jeu d'URL qui se paramètrent dans le fichier web.xml d'inJAC (injac/WEB-INF/web.xml) :

```
<!-- CAS FILTER -->
<filter>
<filter-name>Filtre CAS</filter-name>
<filter-class>edu.yale.its.tp.cas.client.filter.CASFilter</filter-class>
<init-param>
<param-name>edu.yale.its.tp.cas.client.filter.loginUrl</param-name>
<param-value>https://cas.enseeiht.fr:8443/cas/login</param-value>
</init-param>
<init-param>
<param-name>edu.yale.its.tp.cas.client.filter.validateUrl</param-name> <!--service
<param-value>https://cas.enseeiht.fr:8443/cas/serviceValidate</param-value>
</init-param>
<init-param>
<param-name>edu.yale.its.tp.cas.client.filter.proxyCallbackURL</param-name>
<param-value>https://saroumane.enseeiht.fr:8443/injac/CasProxyServlet</param-value>
</init-param>
<init-param>
<param-name>edu.yale.its.tp.cas.client.filter.serverName</param-name>
<param-value>saroumane.enseeiht.fr</param-value>
</init-param>
</filter>
```

```
<filter-mapping>
<filter-name>Filtre CAS</filter-name>
<url-pattern>/injac/auth/*</url-pattern>
</filter-mapping>
<!-- CAS FILTER -->
```

Dans cet exemple "**cas.enseeiht.fr**" est le hostname de l'application CAS, et "**saroumane.enseeiht.fr**" celui où est déployé Injac.

Configuration avancée

- Options de rechargement de Cocoon ou du sitemap

Une url permet forcer le rechargement de Cocoon après par exemple une modification de xconf.conf : `http://localhost:8080/injac?cocoon-reload=true` Note: Chaque fois que je l'ai testée, elle a fait planter la servlet Cocoon. En mode développement ce n'est pas gênant, mais en production, il ne faut pas oublier de le désactiver dans `injac/WEB-INF/web.xml`:

```
<!--
  Allow reinstantiating (reloading) of the cocoon instance. If this is
  set to "yes" or "true", a new cocoon instance can be created using
  the request parameter "cocoon-reload"
.-->
<init-param>
  <param-name>allow-reload</param-name>
  <param-value>no</param-value>
</init-param>
```

En outre, le rechargement du sitemap peut aussi être configuré dans `cocoon.xconf` à la dernière ligne :

```
<sitemap file="sitemap.xmap" reload-method="asynchron" check-reload="yes"/>
```

- Configuration mémoire de Tomcat

Cocoon nécessite un minimum de 128M pour fonctionner correctement, sachant que rien qu'au lancement 75mo sont utilisés, et le maximum reste dépendant du nombre d'utilisateurs. Il faut donc indiquer à la jvm que le servlet container doit se lancer avec les options `-Xms128m -Xmx256`(min et max de mémoire) : Dans le répertoire `{$CATALINA_HOME}/bin`, éditer le fichier `catalina.bat` et/ou `catalina.sh`, et chercher la variable `CATALINA_OPTS`, et lui rajouter les options `-Xms128m -Xmx256` :

```
catalina.bat : set CATALINA_OPTS=-Xms128m -Xmx256m -server
```

```
catalina.sh : CATALINA_OPTS="-Xms128m -Xmx256m $CATALINA_OPTS"
```

Niveau de débogage

Le niveau de débogage peut être modifier en éditant le fichier `WEB-INF\classes\log4j.properties`. Les niveaux disponibles sont : **debug**, **info**, **warn**, **error** et **fatal**, du plus proluxe au plus concis.

Note : il faut également adapter le chemin du fichier de log, à la ligne :

```
log4j.appender.R.File=F:/logs/injac.log
```

Personnalisation

Le moteur de rendu inJAC offre la possibilité d'adapter l'apparence du rendu (notion de *skin*), mais également de personnaliser la langue utilisée, grâce au module I18N utilisé dans le framework Cocoon.

Personnalisation du rendu

La personnalisation se fait à travers deux fichiers associés :

- un fichier de *skin*

Il définit des options du contenu du rendu (liens, logos, recherche).

Emplacement : le sous-répertoire `injac/xml/skins`.

Le fichier `skin.xml.template` sert de modèle auquel se référer pour construire d'autres fichiers.

- une feuille CSS

Elle gère l'aspect graphique du rendu (couleurs, disposition, images...).

Emplacement : le sous-répertoire `injac/stylesheets/xsl`.

Le fichier `skin-esup1.css` peut servir de modèle auquel se référer pour construire d'autres fichiers.

C'est à travers l'édition des métadonnées lors de la création ou de l'administration d'un espace par le **Canal de Stockage** que l'administrateur d'espace choisit le skin :

Le **choix d'un skin** dans le Canal détermine l'utilisation, dans le moteur de rendu, des **fichiers XML et CSS** correspondant, qui doivent impérativement **porter le même nom** (sauf l'extension, bien sûr).

Une liste des skins doit donc être renseignée dans le fichier de configuration du Canal, `properties/injac/injac.xml`, sous la forme :

```
<RENDERINGS>
  <RENDERING
    skinFile="skin-esup1.xml"
    label="ESUP 1"
```

```
</>
<RENDERING
  skinFile="skin-esup2.xml"
  label="ESUP 2"
/>
<RENDERING
  skinFile="skin-esup-doc.xml"
  label="Documentation ESUP"
/>
<RENDERING
  skinFile="skin-valenciennes.xml"
  label="Université de Valenciennes"
/>
<RENDERINGS>
```

- Description du fichier de *skin*

Le skin est un fichier xml qui permet de définir principalement le **contenu** du rendu d'un espace inJac :: les images et liens que l'on veut rajouter, les éléments optionnels tels que recherche, RSS, contenus issus de base de données.

Voici un squelette de fichier skin, montrant les éléments de type contenu :

```
<skin>
  <header> Entête
    <logos/> Image(s)
    <links /> Liens
  </header>
  <navbar/> Barre de navigation
  <navpan/> Panneau de navigation
  <doc/> Document
  <tools> Outils divers : lucene, RSS
    <tool/>
  </tools>
  <footer> Pied-de-page
    <validators> Validateurs CSS, HTML, XHTML
    <validator/>
    </validators>
  </footer>
</skin>
```

- Description du fichier CSS

Le fichier CSS sert à habiller le contenu avec des **éléments de style** : fontes, couleurs, images de fond...

Les classes et identifiants insérés dans le code HTML sont affectés à ces éléments de style par le feuille CSS, et il suffit donc de changer les styles associés à chaque nom de ces classes et identifiants pour personnaliser librement l'apparence.

Voici une liste arborescente montrant l'inclusion de ces noms et leur correspondance avec un élément du contenu. Pour chaque point d'arborescence, le premier mot représente la balise HTML, et le deuxième la classe (.) ou l'identifiant (#) CSS :

- div #page

- div #header (en-tête)
 - div #headerLinks (liens de l'entête)
- div #welcome (bienvenue utilisateur)
- div #login (lien pour l'authentification)
- div #menu
 - div #navbar (barre de navigation)
 - div #sectionMenu
 - div #iNavPan (panneau de navigation)
 - h2 (titre)
 - ul .ispaces (espaces inJAC)
 - ul .idocs (documents inJAC)
 - div #gentools (outils)
 - div .tool
- #iDocRender (zone de rendu centrale des documents)
 - div #iDocToPDF (lien pour la conversion PDF)

Deux skins principaux sont fournis avec le package :

skin-esup1 :

ESUP Portail Site de l'ENSEEIHТ Site de l'INP Portail ESUP ENSEEIHТ S'identifier

inJAC >>DOCUMENTATION >>PRÉSENTATION

Documentation
 ▶ Présentation

Chercher

recherche avancée

Moteur de rendu inJAC

François Jannin

ENSEEIHТ

Historique des versions	
Version 2.3	1 décembre 2005

Résumé

Le moteur de rendu inJAC permet de naviguer parmi les contenus des espaces inJAC créés via le Canal de Stockage, et, en jonction avec le moteur d'indexation, d'effectuer des recherches sur les contenus et méta-données. Il utilise le framework Apache Cocoon ainsi que le moteur d'indexation Lucene.

skin-esup2 :

ESUP Portail Site de l'ENSEEIHТ Site de l'INP Portail ENSEEIHТ S'identifier

inJAC >>Bienvenue

inJAC

- Documentation
- Presentations
- Ressources pédagogiques
- Test
- Theses
- Bienvenue exemple html composite
- Lucene et Cocoon

Bienvenue dans l'espace de publications d'Injac

pages rendues par Cocoon

Le rendu de ces pages est réalisé par Cocoon, moteur de transformations XSLT qui permet une conversion dans un grand nombre de formats (HTML,XML,PDF, etc...).

En haut à droite de chaque document se trouve une icone qui permet de le télécharger au format PDF.

Chercher

recherche avancée

Localisation

Le moteur de rendu inJAC permet un affichage dans différentes langues, et par défaut, l'anglais et le français sont supportés dans cette version.

Cette traduction est assurée par le module i18N [<http://cocoon.apache.org/2.1/userdocs/i18nTransformer.html>] intégré à Cocoon.

Pour sélectionner la langue d'affichage, il suffit d'indiquer une des deux valeurs suivantes pour le paramètre <locale> du fichier injac/conf/iconf.xml :

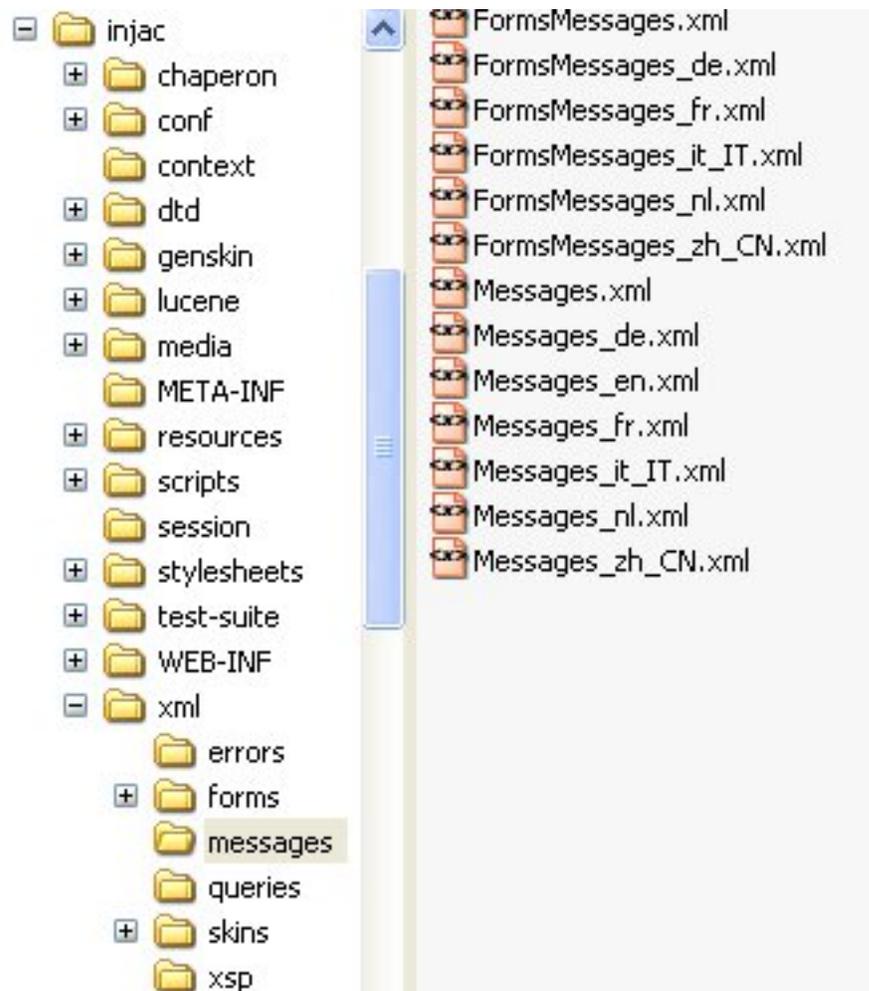
- "en" pour l'anglais

- "fr" pour le français

Les fichiers assurant la traduction se situent dans le répertoire injac/xml/messages.

Le fichier pour le français : messages_fr.xml, pour l'anglais : messages_en.xml.

Remarque : Les clés pour la traduction sont en anglais, aussi le fichier messages_en.xml ne contient-il pas toutes les entrées, car la valeur de la clé est utilisée pour l'affichage par défaut du texte.



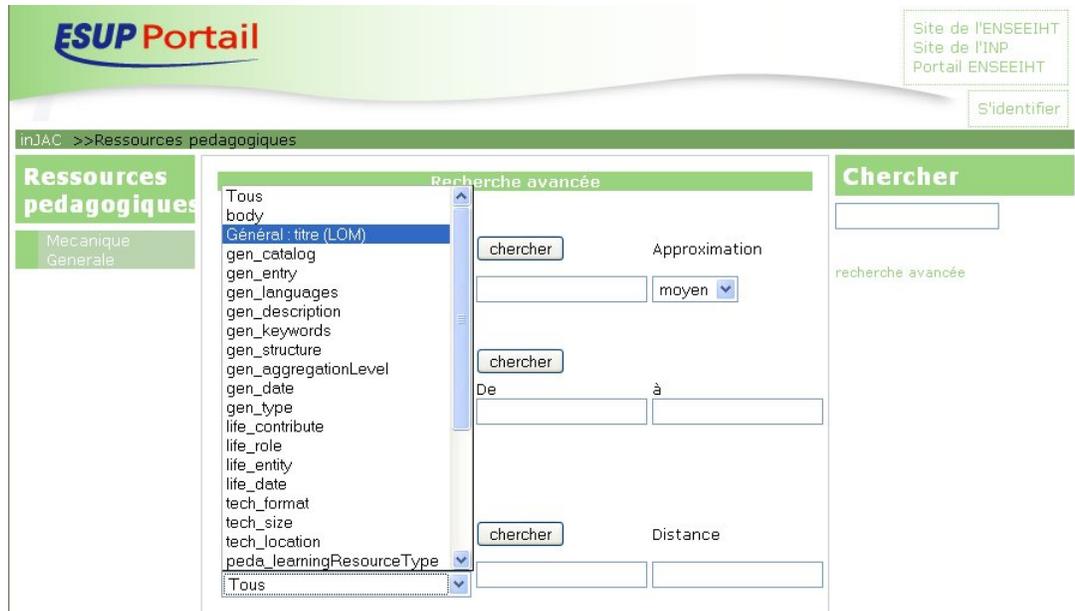
Les fichiers FormsMessages_*.xml sont les fichiers prédéfinis pour le framework de formulaires CForms.

Les personnalisations de la langue concernent deux objectifs différents :

- Compléter les dictionnaires anglais et français en rajoutant, par exemple, la traduction des nom des métadonnées pour la recherche avancée :

```
<!-- METADATA FIELD NAMES -->  
<message key="gen_title">Général : titre (LOM)</message>
```

Cette ligne ajoutée au fichier messages_fr.xml permet de proposer un texte d'affichage différent de celui de la métadonnées WEBDAV, dans la page de recherche avancée :



- Prendre en charge une nouvelle langue

Pour implémenter une nouvelle langue, il faut créer et remplir un fichier messages_**.xml dans le répertoire injac/xml/messages en reprenant la liste des entrées définies dans messages_fr.xml. La valeur à renseigner pour le paramètre locale du fichier iconf.xml doit correspondre au **.

```
<catalogue>
  <message key="org.apache.cocoon.forms.samples.Sex.FEMALE">femenino</message>
  <message key="org.apache.cocoon.forms.samples.Sex.MALE">masculino</message>
  <message key="Welcome">Bienvenida</message>
  <message key="Login">Conexión</message>
```

La valeur des attributs **key** est reprise à l'identique, seule le contenu de la balise **<message>** change.