

Chainedit 3

Création du projet à Partir de Esup-Blank

- [Lien vers la doc Esup](#)
- Le projet est maintenant sur le trunk du SVN de ChainEdit (27/08/2014)
- Le choix est d'utiliser l'archétype web-jsf-servlet
- Toute référence à esup-commons a été retirée (annonce fin du support esup-commons v2), le projet est donc maintenant un projet JSF 2 (View), Spring (Model/Controler), Hibernate / Jaxb (Accès bas et données XML)

Ajout d'une base de donnée embarquée

- A la place d'utiliser certains fichiers XML, il paraît plus judicieux d'utiliser une base embarquée.
- [HSQL](#)
- [Squirrel pour éditer la base](#)
- [Tuto pour brancher Squirrel sur HSQL](#)
- Modifier les accès base : chainedit3-web-servlet/src/main/resources/properties/default.properties → hibernate.connection.jdbc.url
- User par défaut : admin / admin
- [Annotation hibernate pour la validation](#)

Analyse pour la base

[Analyse SI pour faire le MCD car pas d'outil UML fonctionnel](#)

Lancement de l'application sous Tomcat

- [En cas d'erreur java.lang.ClassNotFoundException: org.springframework.web.context.request.RequestContextListener](#)
- Surprise sur le CAS, il faut modifier le Web.xml : CAS Authentication Filter avec>

Et dans le Web.xml de l'application il faut préciser la taille maxi des fichiers :

```
<servlet>
  .....
  <multipart-config>
  <max-file-size>52428800</max-file-size>
  <max-request-size>52428800</max-request-size>
  <file-size-threshold>0</file-size-threshold>
```

```
</multipart-config>  
</servlet>
```

Mise en place de Spring-Security

- A priori plus besoin des classes dans domain.services.auth
- Ni du domain-services-auth.xml
- Ni de la propriété liée dans default.properties et config.properties : auth.bean
- Ni de l'import du auth.xml dans l'applicationContext.xml
- Ni des filtres CAS dans le WEB.xml
- La documentation : [Spring-Security](#)
- Documentation pour le module CAS : [Paramétrage](#)
- Un tuto sur la personnalisation de l'authentification [tuto UserDetailsService](#)
- Il y a 2 modes d'authentification livrés avec Chainedit : CAS et MANUAL (saisie d'un login/password). Il y a donc 2 fichiers application-context-security-cas et un application-context-security-manual. Il n'est pas possible de le passer en paramètre. Il faut donc indiquer le bon nom de fichier dans le import du application-context-security.

Mise en place de Spring-Ldap

- Pour l'accès aux utilisateur dans la page de gestion d'un utilisateur.
- Remplace le LdapService de Esup-commons, la configuration est conservée à l'identique (config.properties) par contre le LDAP.XML est modifié pour utiliser Spring-LDAP.

Zip Zip

[TrueZip](#) pour l'édition des fichiers zip

Intégration i18n

- Le ResourceBundle de Spring est utilisé (appelé msgs en Spring). Un objet ResourceBundleBean (appelé msg en Spring) a été ajouté pour permettre de transmettre les messages aux JSF sans avoir à re-déclarer les bundles dans la partie View JSF (faces-config). Ainsi tous les bundles passent par Spring.
- Les fichiers custom ne doivent pas être utilisés, ils sont à laisser au client pour surcharger les chaînes de messages livrés avec l'application.
- Ajout d'un éditeur de bundles [JExpresso Translation](#) qui permet d'éditer les variables directement depuis la classe où il est utilisé
- Ajout d'un éditeur de bundles [Properties Editor](#) coloration du contenu et gestion de l'Unicode pour un fichier

Spring JPA Repositories

- [Spring data repositories](#)
- [JUnit](#)
- [Aide HQL Hibernate](#)

JSF

- [Tuto JSF](#)
- [Templating JSF](#)
- [Tenir compte des préférence de l'utilisateur concernant son thème \(Skin dans ChainEdit\)](#)
- [Liste des icones Primefaces](#)
- [Showcase Primefaces](#)
- [Layout primefaces](#)
- Un fichier `prime_override.css` a été ajouté pour permettre de surcharger les classes css standard de prime
- [Compléments à JSF Omnifaces](#)

L'éditeur CKEditor fourni avec Primefaces ne peut pas être customisé simplement. Le choix est fait d'intégrer directement l'éditeur :

- [CKEditor 4 1](#) éditeur maximum accepté sur le même écran.
- [Plugin mathjax pour latex](#) dans `fr.js` et `en.js` *button: Math* a été changé en *button: Latex* pour éviter les confusions avec le plugin MathML
- [Plugin mathjax modifié pour gérer du mathml : renommé en mathjaxmathml](#)
- [Wiki ChainEdit sourcesup sur CKEditor](#)

Bean validation

- Il est possible depuis Java EE 6 d'annoter les beans pour ajouter des contraintes sur leurs valeurs (JSR 303) : [spécification Bean Validation 1.1](#)
- [Hibernate Validator](#) spécifie la JSR 349 qui est plus complète : https://docs.jboss.org/hibernate/stable/validator/reference/en-US/html_single/
- [Apache Commons Lang](#) pour faire des validation simple et d'autres utilitaires sur les Beans (equals, hashCode, ..)

Imports dans ChainEdit 3

[Les imports](#)

Preview dans ChainEdit 3

[Les chartes preview](#)

From:

<https://wiki.univ-rennes1.fr/chainedit/> - **chainedit wiki**

Permanent link:

<https://wiki.univ-rennes1.fr/chainedit/doku.php?id=developpeurs:chainedit3>

Last update: **2017/03/30 12:53**

