

Utilisation esup-portlet-mondossierweb

Utilisation

- Pour déployer l'application dans le Tomcat configuré dans les fichiers properties (voir 2em point de l'installation) : target 'deploy'

Important :

- Le Tomcat utilisé sera celui configuré dans le fichier 'properties' (créé précédemment) correspondant au déploiement choisi.
- Dans le cas où le Tomcat utilisé pour uportal est différent de celui où est déployé esup-mondossierweb (notamment en servlet), ne pas oublier de faire attention aux ports utilisés par les deux serveurs, de manière à ce qu'il n'y ait pas de conflit.
- Dans le cas d'un déploiement portlet, ne pas oublier de déclarer la portlet dans le contexte de Tomcat pour pouvoir l'ajouter comme canal d'uportal :

```
<Context path="/esup-mondossierweb" docBase="pathVersTomcat/webapps/esup-mondossierweb" crossContext="true" reloadable="true"/>
```

- Rappelons que pour un déploiement portlet au sein d'Uportal, esup-mondossierweb doit être déployé au même niveau qu' UPortal.
- Dans le cas d'un déploiement portlet, il peut être nécessaire d'ajouter **emptySessionPath="true"** dans la déclaration du ou des 'Connector' dans le server.xml de Tomcat pour que le téléchargement de fichier (pdf, excel) fonctionne.
- Dans le cas d'un déploiement portlet, voici le 'Portlet Définition ID' à mettre lors de la déclaration de la portlet dans le portail (param portlet-guid du web.xml): **esup-mondossierweb.esup-mondossierweb**.
- Pensez également à mettre un 'Channel timeout' assez élevé (15000 par ex) car certains rendus sont (à cause de facelets notamment) assez long à s'afficher.

Voici un exemple de déclaration :

User can modify?	Name	
	Channel Type:	Portlet
	Channel Title:	esup-mondossierweb
	Channel Name:	esup-mondossierweb
	Channel Functional Name:	esup-mondossierweb
	Channel Description:	
	Channel Timeout:	15000 milliseconds
	Channel Secure:	☐
☐	Portlet definition ID	esup-mondossierweb.esup-mondossierweb
	Channel Controls	☐ Editable ☐ Has Help ☐ Has About

- Vous pouvez à tout moment passer du déploiement servlet à portlet (et vice versa) en utilisant les targets appropriées du build.xml (voir 2em point de l'installation).
- Les fichiers de configuration se trouve dans le répertoire 'properties'. Ils permettent de configurer :
 - la connexion à Apogée, le mail d'erreur, ldap, etc : application.properties
 - les urls du WS de l'amue : configUrlServices.properties
 - le logger : logging/ log4j.properties
 - toutes les options d'affichage, de paramétrage et le serveur de photo: monDossierWeb/monDossierWeb.xml

Ils sont détaillés ci-dessous

- Lors de l'identification si l'utilisateur est reconnu en tant qu'étudiant, il est dirigé vers la page d'accueil 'étudiant' de l'application. Sinon, s'il est reconnu en tant qu'utilisateur uportal ou en dernier recours en tant qu'utilisateur Apogée, il est redirigé vers la page d'accueil 'enseignant' de l'application. Dans tout autre cas, l'utilisateur est redirigé vers une page d'erreur lui indiquant qu'il n'est pas autorisé à accéder à l'application.
- En déploiement servlet, vous pouvez vous déconnecter en cliquant sur 

Les fichiers de configuration

Voici la liste des options configurables grâce aux fichiers du répertoire 'properties' :

- application.properties:
 - le type de connexion à la BDD (Pool tomcat ou pas)
 - les identifiants de connexion à la base
 - le serveur smtp
 - l'email recevant les erreurs
 - l'email de contact
 - les identifiants de connexion au ldap
 - l'url du WS uportal
 - les classes de récupération du login en fonction du codind et inversement.
- configUrlServices.properties :
 - les urls d'accès au WS de l'AMUE.
- log4j.properties :
 - fichier de configuration typique log4j (sortie des logs, fichier de log etc...)
- monDossierWeb.xml :
 - activation de l'application (accessibilité pouvant s'avérer utile via JMX).
 - activation de la partie enseignant
 - activation de la partie étudiant
 - logger les connexions sur l'appli (et dans un fichier de log séparé) : voir monDossierWeb.xml et log4j.properties
 - activer le trombinoscope
 - afficher un nombre fixe d'étudiants par page affichée du trombinoscope
 - activation du lien 'nous contacter'
 - l'e-mail du lien 'nous contacter'. Ex : `scolarite@univ-nancy2.fr`
 - l'extension au login pour l'e-mail des étudiants. Ex : `@univ-nancy2.fr`
 - le message de maintenance.
 - le message d'erreur de connexion
 - le message d'interdiction d'accès
 - le message lors de la visualisation des notes non définitives
 - Affichage des inscriptions de l'étudiant.
 - Affichage du calendrier.
 - Possibilité de générer le calendrier standard en pdf
 - Possibilité d'afficher la mention dans la page des notes et résultats
 - affichage du calendrierEtape (implémenté par Rennes1) à la place du calendrier standard.
 - affichage du calendrier de rentrée (implémenté par Rennes1)
 - Affichage des notes de l'étudiant.
 - Possibilité de trier les notes sur le témoin fictif d'Apogée (on affichera que les ELP dont le témoin fictif d'Apogee est égal à `tem_fictif`)
 - L'utilisation d'un `dataScroller` pour afficher la liste des étudiants et le nombre d'étudiants affichés par page
 - Le nombre d'étudiants par page quand on utilise le `dataScroller`
 - récupération par le WS du login et de l'email dans APOGEE(`pkb_annu` et `TEM_ACC_ANNUAIRE` doivent être actifs)
 - activer la modification des l'adresses pour l'étudiant
 - La source de Résultat lors de la récupération des notes avec le WS `amue.'Apogee'` par défaut.
 - Témoin état de délibération des notes vues par l'étudiant.
 - Témoin état de délibération des notes vues par les enseignants.
 - Possibilité d'appliquer la descendance des témoins du semestre : on affiche les notes que si le semestre père est à T (utile pour UEs partagées) (paramètre `temNotesEtuSem`)
 - Affichage du rang aux Diplômes et Epreuves.
 - Affichage des crédits ECTS sur le détail des notes.
 - Activer la génération des notes en pdf pour l'étudiant.
 - Insérer un filigrane dans le PDF des notes.
 - Activer la génération des certificats de scolarité en pdf.
 - Autoriser ou non les personnels à imprimer les certificats de scolarité.
 - Le signataire des certificats de scolarité.
 - Le lieu d'édition des certificats de scolarité.
 - Activer l'affichage du logo dans les certificats de scolarité.
 - Liste des codes de types de diplômes pour lesquels la génération de certificat est désactivée.
 - Possibilité d'interdire l'édition de certificat de scolarité en fonction du profil étudiant (paramètre `certScolProfilDesactive`)
 - Possibilité d'interdire l'édition de certificat de scolarité en fonction du code CGE de l'étape (paramètre `certScolCGEDesactive`)
 - Possibilité d'interdire l'édition de certificat de scolarité en fonction du code composante de l'étape (paramètre `certScolCmpDesactive`)
 - Ajout du tampon sur le certificat de scolarité (paramètre `certScolTampon`)
 - Ajout de bannières supérieure et inférieure sur le certificat de scolarité (si logo défini, affichage du logo plutôt que de la bannière supérieure) (paramètres `certScolHeaderUniv` et `certScolFooter`)
 - Indiquer une url pour insérer une image dans les pdf.
 - La résolution en dpi de l'image de la signature dans le certificat de scolarité.
 - Activer l'affichage de la bannière en servlet (utile à 'false' pour une iframe dans uportal).
 - Affichage du numéro de place dans le calendrier des examens
 - Message complémentaire concernant le calendrier des examens
 - Message complémentaires concernant le calendrier de rentrée
 - l'identifiant (login) de l'étudiant pour lequel on souhaite se faire passer pour un test.
 - l'identifiant (login) qui souhaite se faire passer pour l'étudiant si dessus.
 - donner une liste de login à exclure de l'application
 - Configurer le bean Security. (le type ldap d'un étudiant, l'attribut qui stocke le `codEtu` dans le ldap, les groupes uportal autorisés à accéder à l'application, `loginApogee` pour activer/désactiver la recherche du login utilisateur dans Apogee.)
 - Configurer le bean Photo. (Empty, Nancy2)
- Custom_fr.properties (dans `properties/i18n/bundles`) :
 - toutes les chaînes de caractères présentes dans les documents générés par la portlet.

Les beans interchangeables

Certains beans ont été conçus de manière à être facilement remplacés si besoin est. C'est à dire que vous pouvez créer votre propre bean si celui proposé ne convient pas à votre environnement. Il suffit pour cela de créer une nouvelle classe java implémentant l'interface du bean d'origine, puis de modifier la description du bean ('class' et pourquoi pas 'property') dans le fichier XML correspondant.

Ces beans sont les suivants:

identifiant	fichier XML (répertoire 'properties')	rôle	interface à implémenter
photo	monDossierWeb.xml	récupération des photos des étudiants	IPhoto
security	monDossierWeb.xml	Gestion de l'accès à l'application	ISecurity
etudiantManager	monDossierWeb.xml	récupération des informations du dossier d'un étudiant. A l'origine effectué à l'aide du WS de l'AMUE	IEtudiant
daoService	dao.xml	récupération des informations de la partie enseignant de l'application. A l'origine en utilisant iBATIS	IDaoService
daoCodeLoginEtudiant	application.properties (dao.xml)	récupération du cod_ind à partir du login d'un étudiant.	IDaoCodeLoginEtudiant
daoLoginCodeEtudiant	application.properties (dao.xml)	récupération du login à partir du cod_ind d'un étudiant	IDaoLoginCodeEtudiant

A noter que la récupération du login en fonction du codInd (et vice versa) se fait, de base, via une table MEL_LOGIN propre à l'université NANCY2. C'est pourquoi les universités ne disposant pas de cette table doivent implémenter leur propre récupération d'information à ce niveau sous peine d'une erreur oracle : **ORA-00942: table or view does not exist**. Les beans interchangeables concernés sont : daoCodeLoginEtudiant et surtout daoLoginCodeEtudiant. En effet, daoCodeLoginEtudiant tente une récupération via LDAP, avant d'aller chercher les infos dans la table MEL_LOGIN.

JMX

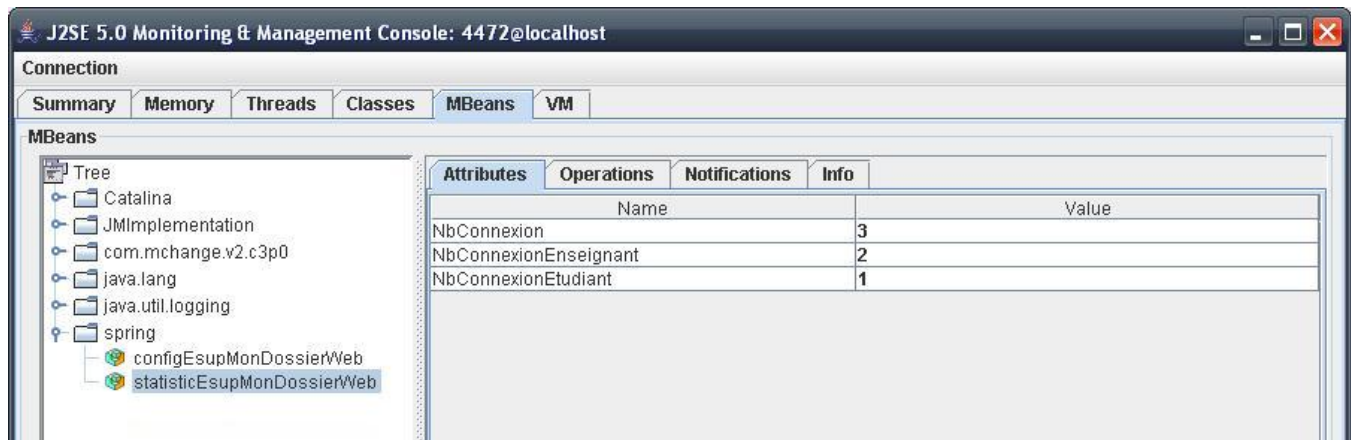
Grâce a JMX vous pouvez consulter et modifier certains beans en temps réel.

Pour pouvoir utiliser jmx vous pouvez soit : utiliser la target 'start' du build.xml pour lancer Tomcat, soit lancer Tomcat en indiquant : **-Dcom.sun.management.jmxremote=true** dans la ligne de commande ou en indiquant **SET CATALINA_OPTS="-Dcom.sun.management.jmxremote"** dans votre fichier .cmd qui lance Tomcat.

Une fois Tomcat lancé vous pouvez utiliser la **jconsole** (en rentrant par exemple jconsole dans l'invité de commande windows).

Une fois dans la jconsole vous pouvez aller dans l'onglet **MBeans** et dans le dossier **Spring** ; vous y trouverez les beans configEsupMonDossierWeb et statisticEsupMonDossierWeb vous permettant respectivement de modifier les paramètres et de consulter les statistiques de l'application en temps réel à l'aide du bouton 'Refresh'.

Exemple:



FAQ

-J'ai un problème d'accent dans les pdf générés. J'ai des '?' à la place des accent

En cas de problème d'accents dans les pdf générés cela peut venir de la locale. Il faut une locale compatible avec le canal. En général, cette ligne dans le démarrage du tomcat solutionne le problème :
export LANG=fr_FR@euro

-Comment utiliser le trombinoscope?

Pour la récupération des photos et des trombinoscopes, nous utilisons un serveur de photo spécifique à Nancy2. C'est une application interne que nous ne distribuons pas ; chacun doit donc faire sa propre implémentation. Nous avons laissé nos beans d'appel à ce serveur de photos dans les sources distribuées pour qu'ils puissent servir d'exemples. Pour information sur le principe de fonctionnement, il délivre des tickets autorisant l'accès à des urls pointant vers les photos des étudiants

-J'ai un problème à l'appel des WS de l'AMUE.

Il faut bien penser à renseigner toutes les infos dans configUrlServices.properties ; même les urls ssl si on ne s'en sert pas, ainsi que WS_KEY_STORE_PATH etc.

-J'ai une erreur de type 'Cannot create JDBC driver' :

Essayez, en mettant le driver Oracle dans le répertoire common de Tomcat.

-Je stocke le cod_ind dans le ldap et pas le cod_etu. comment faire ?

Il suffit d'indiquer 'codetu.login=DaoCodeLoginEtudiantImplNancy2' dans application.properties. Cette classe va chercher le codInd dans le ldap puis demande aux WS le cod_etu correspondant.

-Je n'utilise pas les WS, et certaines infos sont manquantes, ou certaines options ne sont pas prises en compte

Le dossier étudiant en version requêtes SQL n'est plus maintenu sur les dernières versions. Pour disposer de toutes fonctionnalités il faut impérativement passer par les WS.

-Quelles sont les règles d'affichage des notes ?

Les règles de gestion de l'affichage des notes (les témoins dépendent de la configuration de l'application dans monDossierWeb.xml) :

- Les enseignants voient toutes les notes (si on configure le témoin à "AET" pour les enseignants). Que l'état de délibération soit A, E ou T
- Les étudiants voient les notes dont l'état de délibération est à T (si on configure le témoin à "T" pour les étudiants). A noter qu'ils peuvent voir la note à une épreuve si l'état de délibération de celle-ci est à T même si l'ELP correspondant n'est pas dans un état de délibération à T. Ils peuvent également voir les notes aux épreuves qui sont à l'état A ou E si la case "modalité de contrôle des connaissances" n'est pas cochée dans le menu épreuve du domaine MCC.

-Quelles sont les règles d'affichage du lien de téléchargement du certificat de scolarité (onglet scolarité)?

- certificatScolaritéPDF doit être à true dans monDossierWeb.xml
- certScolAutorisePersonnel doit être à true dans monDossierWeb.xml si l'utilisateur est un enseignant.
- L'année de l'inscription doit être l'année en cours dans APOGEE
- Le code type du diplôme ne doit pas être dans certScolTypDiplomeDesactive (dans monDossierWeb.xml).
- L'inscription doit être en règle (payée).

-Je suis en uPortal inférieur à v3 et j'ai une erreur : javax.portlet.PortletException: Unable to find dispatcher for context: null

Il faut, dans le web.xml, décommenter :

```
<url-pattern>/esup-mondossierweb/*</url-pattern>  
et commenter  
<url-pattern>/PlutoInvoker/esup-mondossierweb</url-pattern>
```

-J'ai une erreur *GRAVE*: "Servlet.service()" pour la servlet jsp a généré une exception java.lang.LinkageError: loader constraint violation: when resolving interface method :

Il s'agit d'un problème de compatibilité de jar avec Tomcat6. Vous pouvez essayer de mettre à jour les librairies el-impl et el-api comme suit :

- suppression de el-api.jar dans webapp/WEB_INF/lib
- suppression de el-impl-1.0.jar dans webapp/WEB_INF/lib
- ajout de el-impl-2.2.jar dans webapp/WEB_INF/lib en le renommant 'el-impl.jar' ([download](#))
- ajout de el-api-2.2.jar dans **tomcat/lib** ([download](#))
- lancer la tâche Ant : clean
- lancer la tâche Ant : deploy