

3.2.1. Configuration de config.properties



Explication préalable

Le fichier default.properties contient les configurations par défaut de l'application.
Ainsi si on rajoute une propriété déjà existante dans le fichier config.properties, celle-ci écrasera celle du fichier default.properties.
La configuration du fichier config.properties consiste à modifier les valeurs présentes dans le fichier ou parfois à rajouter une entrée définie dans default.properties

Table des matières

- Configuration de l'accès aux feuilles de style CSS
- Configuration d'hibernate
 - Configuration en JDBC
 - Configuration en JNDI
- Configuration de l'authentification
- Configuration de la gestion d'exceptions
- Configuration des données d'initialisation
 - Configurations globales de l'application
 - Récupération d'informations depuis Apogée
- Configuration des tâches planifiées
- Configuration de la connexion au LDAP
 - Paramètres de connexion au LDAP
 - Paramètres de recherche dans LDAP
- Configuration de la connexion au serveur SMTP
- Configuration du module de génération de PDF
- Configuration des services utilisant Orbeon
- Configuration de la génération d'URL
- Configuration de la connexion aux webservices ESUP-WS-SI
- Configuration du fuseau horaire
- Configuration du Logging de l'application
- Configuration de la gestion du cache
- Configuration des URL des IAWeb
- Configuration de l'accès à un calendrier de candidatures externe

Configuration de l'accès aux feuilles de style CSS

```
mediaPath.servlet=/esup-opiR1/media
```

Dans le fichier default.properties, ce paramètre est défini à /esup-opiR1/media. Il définit le chemin absolu d'accès aux feuilles de style CSS.

Selon si votre accès diffère, il faut rajouter cette ligne dans le fichier config.properties, par exemple :

- pour l'url : <http://localhost:8080/> --> mediaPath.servlet=/media
- pour l'url : <http://localhost:8080/ecandidat/> --> mediaPath.servlet=/ecandidat/media

Configuration d'hibernate

Configuration en JDBC

Si vous vous connectez à votre base avec JDBC, configurez les données ci-dessous :

```
hibernate.connection.driver_class
```

C'est la classe hibernate pour communiquer avec la base de données

- pour MySQL (par défaut), utiliser com.mysql.jdbc.Driver
- pour Oracle, utiliser oracle.jdbc.OracleDriver (choisir le pilote pour la base appropriée)

```
hibernate.dialect
```

Le dialecte à utiliser pour interroger la base de données (InnoDB ou autres)

- pour MySQL (par défaut), utiliser `org.hibernate.dialect.MySQLInnoDBDialect`
- pour MySQL (version 5 ou supérieure), utiliser `org.hibernate.dialect.MySQL5InnoDBDialect`
- pour Oracle, utiliser `org.hibernate.dialect.Oracle10gDialect`

```
hibernate.connection.jdbc.url
```

Il s'agit de l'url à laquelle se trouvent les fichiers de la base de données

```
hibernate.connection.jdbc.username
```

Le nom d'utilisateur à fournir pour que l'application puisse se connecter à la base de données

```
hibernate.connection.jdbc.password
```

Le mot de passe associé au nom d'utilisateur précité

```
hibernate.useJndi=false
```

placé à false car il s'agit d'une connexion directe JDBC

Configuration en JNDI

Si vous vous connectez à votre base via un pool de connexion JNDI, il faut configurer les données de cette manière :

```
hibernate.useJndi=true
```

placé à true, la connexion va s'effectuer via une pool de connexion JNDI (installée sur le tomcat)

```
hibernate.connection.jndi.datasource=jdbc/opi
```

Il s'agit de l'url de la source de données employée par JNDI. (par exemple jdbc/opi)

Il faut configurer le fichier `server.xml` de votre tomcat pour définir la ressource JNDI et les paramètres de connexion à la base dans le contexte de votre application :

```
<Context path="/candidatures" docBase="{chemin absolu de l'application}" reloadable="false" debug="0">
  <Resource name="jdbc/opi" auth="Container"
    type="javax.sql.DataSource"
    username="{db_user}" password="{db_password}"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://{hôte de la base}:{port}/{nom de la base}"
    validationQuery="SELECT 1 FROM dual"
    maxActive="100" maxIdle="10" maxWait="10000"
    removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true" />
</Context>
```

pour la propriété `driverClassName` :

- pour MySQL, utiliser `com.mysql.jdbc.Driver`
- pour Oracle, utiliser `oracle.jdbc.OracleDriver`

Configuration de l'authentification

```
cas.url
```

Tout simplement l'URL du CAS à employer pour s'authentifier. Attention cependant à ne pas laisser traîner de slash à la fin de l'URL.

Configuration de la gestion d'exceptions

```
exceptionHandling.logLevel
```

Le niveau de logging (une valeur parmi TRACE, DEBUG, INFO, WARN, ERROR ou FATAL) :

- **TRACE** note absolument tout ce qui se passe dans l'application
- **DEBUG** note toutes les informations nécessaires au débogage
- **INFO** note l'essentiel de l'activité de l'application
- **WARN** note uniquement les erreurs et avertissements
- **ERROR** ne note que les erreurs
- **FATAL** ne note que les erreurs critiques

```
exceptionHandling.email
```

Il s'agit de l'adresse e-mail à laquelle les rapports d'exceptions seront envoyés (donc de préférence l'adresse de la personne chargée d'entretenir l'application).

Configuration des données d'initialisation

Configurations globales de l'application

```
init.firstAdministratorId
```

L'identifiant LDAP du premier administrateur de l'application. Il s'agit de son login de connexion. Celui-ci est ensuite récupéré dans LDAP lors de la tâche ant init-data et sert à établir les informations nécessaires à la création du premier administrateur.

```
init.codEtbInt
```

Il s'agit du code de **votre** établissement tel qu'il est connu dans Apogee.

```
init.typEtbUniv=00
```

Il s'agit du code du type d'établissement "Université" tel que défini dans Apogée.

Vous pouvez le trouver dans la partie Référentiel, menu Etablissement / Environnement / Type d'établissement.

Par défaut, le code pour le type d'établissement "Université" est 00.

```
init.codDacEtr
```

Le code diplôme autre cursus (DAC) employé dans Apogee pour identifier les cursus des étudiants étrangers

```
init.codStudent.regex=e([0-9]{7})
```

Substitution du premier caractère du code étudiant par « e » pour conformité avec les caractéristiques du login étudiant de l'établissement : « e »+7 dernier caractères du code étudiant Apogée

```
init.codStudent.pattern=2\$1
```

Ajouter s'il n'y est pas (demandé à l'initialisation) --->substitution du « e » contenu dans le login par le caractère « 2 » pour compléter le code étudiant. Cet argument est utilisé lors de l'appel de l'application de type ent; après authentification via le cas étudiant

```
init.onlyTerminaux
```

Booléen à renseigner selon :

- true si l'on veut uniquement récupérer le dernier diplôme en date des candidats
- false si l'on veut récupérer l'intégralité de leurs diplômes

```
init.obtentionDip
```

Booléen à renseigner selon :

- true si l'on ne prend en compte que les diplômes obtenus
- false si l'on prend aussi en compte le niveau avéré

```
init.context
```

le contexte interne à l'application (utilisé pour générer des urls eg. Faq et contact)

```
init.opiWebService.bean
```

Nom de bean à renseigner pour le déversement de données dans Apogée

- **opiWebService**, si l'on veut avoir un déversement des données dans Apogee
- **voidOpiWebService**, si l'on ne souhaite pas déverser les données dans Apogee.

```
init.cmi.manager=false
```

Ce paramètre s'il est à true permet d'ajouter un Responsable de commission (nom, adresse mail et téléphone) lors de l'ajout ou de la modification d'une commission.

Ces données sont ensuite affichées lorsque l'étudiant consulte le détail d'une formation depuis sa page d'accueil.

```
init.allViewPJ=true
```

Booléen à renseigner selon :

- true : les gestionnaires voient toutes les pièces justificatives
- false pour les afficher par droit



Lorsqu'on crée une commission, l'application lui génère automatique un calendrier avec des valeurs préremplies par défaut.

Le code du calendrier est généré à partir d'un préfixe par défaut et du code de la commission.

Le libellé du calendrier est généré à partir d'un préfixe par défaut et du libellé de la commission.

```
init.dateBackDefault=21/06/2010
```

Date de retour de dossier par défaut.

```
init.prefixCodCalCmi=C
```

Préfixe du code par défaut.

```
init.prefixLibCalCmi=Calendrier de
```

Préfixe du libellé par défaut.

Récupération d'informations depuis Apogée

```
init.recupCursus=true
```

Booléen à renseigner selon :

- true, on récupère les cursus des étudiants de l'établissement depuis Apogée lors de la création du dossier
- false sinon

```
init.recupBac=true
```

Booléen à renseigner selon :

- true, on récupère les baccalauréats des étudiants de l'établissement depuis Apogée lors de la création du dossier
- false sinon

```
init.recupInfos=true
```

Booléen à renseigner selon :

- true, on récupère les informations personnelles depuis Apogée lors de la création du dossier
- false sinon

Configuration des tâches planifiées

```
cron.archiveTask=0 30 0 * * ?
```

Valeur configuration l'exécution de la tâche cron d'archivage par le module Quartz (par défaut tous les jours à 0h30)

Pour voir la syntaxe : <http://wiki.opensymphony.com/display/QRTZ1/CronTriggers+Tutorial>

Pour ne pas les lancer, mettre 0 en dernier (* * * * ? 0)

Configuration de la connexion au LDAP

Paramètres de connexion au LDAP

L'application se base sur l'annuaire LDAP pour récupérer des informations lors de la création de gestionnaires et membre de commissions

```
ldap.url
```

L'URL du serveur LDAP : ldaps://ldap.univ-ubs.fr:xxx

```
ldap.userName
```

Le nom d'utilisateur employé pour se connecter à LDAP =cn=candidature,ou=admin,dc=univ-ubs,dc=fr

```
ldap.password
```

Le mot de passe associé au nom d'utilisateur

```
ldap.base
```

La base de recherche de LDAP, par exemple ldap.base=dc=univ-ubs,dc=fr

```
ldap.dnSubPath=ou=people
```

Accès à la branche people pour la recherche

Paramètres de recherche dans LDAP

On utilise des librairies Esup Commons pour faire des recherches dans l'annuaire LDAP.

Voici les différents attributs servant à la recherche :

```
ldap.uidAttribute=uid  
  
ldap.displayNameAttribute=displayName  
  
ldap.emailAttribute=mail  
  
ldap.searchAttribute=cn  
  
ldap.eduPersonPrincipalNameAttribute=eduPersonPrincipalName  
  
ldap.prenomAttribute=givenName  
  
ldap.filterPers=objectClass=person  
  
ldap.nomUsuelAttribute=sn
```

Voici les différents attributs remontés lors d'une requête :

```
ldap.attributes=cn,displayName,mail,sn,givenName,eduPersonPrincipalName  
ldap.searchDisplayedAttributes=cn,displayName,employeeType,department
```

Configuration de la connexion au serveur SMTP

```
smtp.host
```

Le nom de l'hôte ou l'adresse IP du serveur SMTP utilisé pour l'envoi des e-mails

```
smtp.port
```

Le port employé par ledit serveur SMTP

```
smtp.user
```

Le nom d'utilisateur à employer pour se connecter au serveur SMTP. Laisser vide pour une connexion anonyme.

```
smtp.password
```

Le mot de passe à employer pour se connecter au serveur SMTP. Laisser vide pour une connexion anonyme.

```
smtp.interceptAll
```

true en test, false en production. Définit s'il faut intercepter tous les e-mails envoyés ou non

```
smtp.fromEmail
```

L'adresse e-mail apparaissant dans le champ « De : » des e-mails envoyés par l'application. Fonctionne très bien même si l'adresse n'existe pas

```
smtp.fromName
```

Le nom apparaissant dans le champ « De : » des e-mails envoyés par l'application.

```
smtp.interceptEmail
```

l'adresse e-mail vers laquelle rediriger les e-mails interceptés (de préférence celle de la personne chargée de mettre le programme en place)

```
smtp.interceptName
```

Le nom à afficher dans le champ « pour : » des e-mails interceptés (complètement optionnel)

Configuration du module de génération de PDF

```
xsl_xml.path
```

L'emplacement des fichiers xml et xsl utilisés pour générer les fichiers PDF (/var/lib/tomcat/candidatures/webapps/candidatures/media/xml-xsl/)

Configuration des services utilisant Orbeon

L'application embarque une application Orbeon pour la gestion de formulaires dynamiques. L'application crée, lit et supprime ces formulaires en se connectant directement à Orbeon par les paramètres suivants :

```
orbeon.url
```

L'URL de l'application Orbeon. Il suffit de saisir le nom de votre application. Cette URL sera suffixé dans l'application des paramètres permettant l'accès aux formulaires d'Orbeon

```
orbeon.username
```

Nom d'utilisateur à fournir pour utiliser orbeon

```
orbeon.password
```

Mot de passe à fournir pour utiliser orbeon

```
init.orbeon.uri=xmldb:exist://localhost/esup-opiRl/exist/xmlrpc
```

Celui-ci permet de définir le chemin de la base exist dédiée à Orbeon , c'est utilisé par l'application pour accéder aux données Orbeon lors de l'extraction des formulaires complémentaires

Configuration de la génération d'URL



La partie génération d'URL est utilisé par les développeurs et n'est donc pas à prendre en compte pour vos configuration

```
urlGeneration.bean
```

Propriété utilisée par les développeurs. Ne pas en tenir compte.

Le bean à utiliser pour générer les URLs.

- ServletUrlGenerator pour un déploiement en servlet
- uportalUrlGenerator pour un déploiement en portlet

```
urlGeneration.casLoginUrl
```

Propriété utilisée par les développeurs. Ne pas en tenir compte.

L'URL de login du serveur CAS

```
urlGeneration.servletLoginUrl=http://example.domain.edu/stylesheets/cas/accueil.faces
```


Propriété non utilisée. Ne pas en tenir compte.

```
urlGeneration.servletGuestUrl
```

Propriété utilisée par les développeurs. Ne pas en tenir compte.

L'URL de login pour les utilisateurs authentifiés par e-mail. À priori, il suffit de remplacer « exemple.domain.edu » par l'URL d'accès à l'application

```
urlGeneration.uportalFunctionnalName  
urlGeneration.uportalLoginUrl  
urlGeneration.uportalGuestUrl
```

Propriétés utilisées par les développeurs. Ne pas en tenir compte.

Configuration de la connexion aux webservices ESUP-WS-SI

```
remote.flushCache.servers
```

La fonctionnalité de vidage du cache des webservices n'étant pas opérationnelle depuis l'application, ne pas prendre en compte.

Voici la liste des paramètres à configurer pour connecter l'application avec les **webservices esup-ws-si** :

```
remote.referentiel.url=_url_webservices_esup_ws_si_.fr/xfire/ReadReferentiel?WSDL  
remote.enseignement.url=_url_webservices_esup_ws_si_/xfire/ReadEnseignement?WSDL  
remote.insertLaisserPasser.url=_url_webservices_esup_ws_si_/xfire/InsertLaisserPasser?WSDL  
remote.etudiant.url=_url_webservices_esup_ws_si_/xfire/ReadEtudiant?WSDL
```

Configuration du fuseau horaire

```
timezone=Europe/Paris
```

Définit la propriété timezone à travers l'application.

Configuration du Logging de l'application

```
log.level=ERROR
```

Définit le niveau de log (ERROR, INFO ou DEBUG) qui sera écrit.

```
log.output=logfile
```

Définit le moyen d'écriture des logs (stdout pour la console, logfile pour un fichier décrit ci-après)

```
log.file=/data/logs/ecandidat
```

Définit l'endroit de stockage des fichiers de logs

Configuration de la gestion du cache

```
cache.path=/data/cache/ecandidat
```

espace sur lequel sera sauvegardé le cache en cas de dépassement du cache en mémoire vive

Configuration des URL des IAWeb

Lorsque le candidat confirme son vœu, on l'invite à accéder au site d'inscription administratif. Plusieurs cas sont alors possible :

- soit l'inscription administrative par le web est ouverte pour l'étape
 - le candidat fait sa première inscription à l'établissement et on lui donne un accès direct à l'IA primo via son numéro de dossier OPI et sa date de naissance.
 - le candidat se réinscrit dans l'établissement. Il passe par l'ENT puis accède à son onglet d'Inscription Administrative par le web.
 - le candidat était inscrit dans l'établissement mais est en interruption d'étude et ne peut donc pas accéder à son ENT. On lui donne un accès direct via son code étudiant et sa date de naissance au site d'Inscription Administrative par le web.
- soit l'inscription administrative par le web est fermée. Dans ce cas, on propose soit de se rendre directement au service de scolarité, soit d'utiliser le module de prise de rendez vous s'il a été configuré.

```
confirmation.IAWeb.addressIAPrimo=http://_url_site_web_ia_primo_/extern_authent.jsf
```

URL donnant l'accès direct à l'IA primo. L'URL sera suffixée par l'application des paramètres identifiants le candidat par son numéro de dossier OPI et sa date de naissance.

```
confirmation.IAWeb.addressIAReinss=http://_url_site_web_ia_/etapes/identification.do
```

URL donnant l'accès direct au site d'Inscription Administrative par le web. L'URL sera suffixée par l'application des paramètres identifiants le candidat par son numéro d'étudiant et sa date de naissance.

Cette URL est utilisé uniquement dans le cas où l'étudiant est détecté en interruption d'études. Dans l'autre cas, il se verra proposer l'accès à son ENT.

Configuration de l'accès à un calendrier de candidatures externe

```
URL.CalendrierDeCandidature
```

URL affichée si renseignée dans la page "Consulter les formations", page accessible depuis la page d'Accueil.