

# 5. Configuration

- [Pré-requis](#)
- [AdminView](#)
  - [Paramètres de l'application](#)
    - [Le certificat de scolarité](#)
  - [Swap utilisateur](#)
- [Accès à l'application](#)

## Pré-requis

Au premier lancement de l'application, les tables de la base de données MySQL propre à l'application sont créées.

 La table 'schema\_version' indique la version de la base. La base se mettra à jour automatiquement lors des futures livraisons. On retrouvera donc ces trace dans la table 'schema\_version'.

Il faut ensuite ajouter le ou les logins administrateurs dans la table 'administrateurs' de la base.

### Exemple

```
INSERT INTO administrateurs(LOGIN) VALUES ('toto01');
```

Ces utilisateurs ont alors accès à la vue 'admin' (ou '**AdminView**') de l'application permettant de configurer les paramètres 'légers' de l'application.

Contrairement aux paramètres présents dans context.xml/server.xml (cf [ICI](#)), ils ne sont pas indispensables au démarrage de l'application (tomcat) et sont ainsi modifiables à la volée.

## AdminView

A l'installation de l'application, il faut procéder impérativement au paramétrage de l'application.

Pour cela il faut se rendre à l'url suivante avec un compte dont le login est défini dans la table 'administrateurs' (cf pré-requis) : **app.url/#!adminView**

Plusieurs onglets/sections sont proposés :



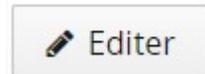
### **NB :**

- Les paramètres de la vue adminView sont, contrairement aux paramètres du context.xml, pris en compte à la volée. Il n'est donc pas nécessaire de redémarrer l'application.
- L'onglet "Swap utilisateur" permet de substituer un login à un autre pendant une durée d'une heure.

## Paramètres de l'application

Les paramètres sont listés dans une table indiquant :

- l'identifiant du paramètre (permet éventuellement de faire le lien avec le paramétrage effectué dans la v2 lors de la migration en v3)
- sa description donnant une indication sur son rôle
- sa valeur



Vous pouvez modifier un paramètre en double-cliquant dessus ou en le sélectionnant puis en cliquant sur

Parmi ces paramètres on retrouve notamment :

- la possibilité de mettre en maintenance l'application dans l'onglet Administration.

## Le certificat de scolarité

L'édition du certificat de scolarité étant une fonctionnalité très utilisée et fortement paramétrable, nous allons voir, pour ce cas précis, les paramètres applicatifs qui rentrent en compte.

Le certificat de scolarité est proposé si les conditions suivantes sont réalisées :

- certificatScolaritePDF= true
- l'utilisateur est un étudiant OU certScolAutorisePersonnel = true
- certificatScolariteTouteAnnee = true OU l'inscription porte sur l'année en cours dans Apogée
- le diplôme ne fait pas partie de la liste des diplômes bloqués (listeCertScolTypDiplomeDesactive)
- si l'utilisateur est un étudiant, l'inscription doit être en règle (payée)
- le témoin de validité du dossier d'inscription doit être égal à 'O' sauf si certificatScolariteDossierNonValide=true
- si l'utilisateur est un étudiant et que certificatScolaritePiecesNonValidees = false, toutes les PJ de l'inscription doivent être valides dans Apogée
- le profil de l'étudiant ne fait pas partie de la liste des profil non autorisés (listeCertScolProfilDesactive)
- le centre de gestion ne fait pas partie de la liste des cge non autorisés (listeCertScolCGEDesactive)
- la composante ne fait pas partie de la liste des composantes non autorisées (listeCertScolCmpDesactive)
- le statut de l'étudiant ne fait pas partie de la liste des statuts non autorisés (listeCertScolStatutDesactive)
- si l'utilisateur est un étudiant et que certificatScolariteCarteEditee=true, la carte doit être indiquée comme éditée dans Apogée

Pour pouvoir ajouter la signature au certificat, il faut également :

- dans context.xml, paramètre "clefApogeeDecryptBlob" : Indiquer la clé Apogée permettant de décrypter les blobs.
- dans l'adminView, onglet "Certificat de scolarité", paramètre "certScolCodeSignataire" : Indiquer le code du signataire



La signature est récupérée dans la table SIGN\_TAMP\_DIGITALISE d'Apogée via le code signataire (COD\_SIG) renseigné dans l'adminView. Elle est décryptée à l'aide de la fonction "decryptLob" du package PKB\_CRY1 en utilisant la clef indiquée dans context.xml (clefApogeeDecryptBlob).

## Swap utilisateur

Tous les "swap" précédemment utilisés sont listés dans l'onglet "Swap utilisateur".

On peut ajouter ou éditer un swap (pour attribuer un nouveau login cible ou modifier la date et l'heure de départ par exemple).

Comme pour les paramètres applicatifs, ce changement est pris en compte à la volée.



Attention, s'il n'est pas nécessaire de redémarrer l'application, il faut cependant penser à se déconnecter/reconnecter si l'utilisateur correspondant au login source était déjà identifié sur l'application.

## Accès à l'application

Une fois l'application déployée elle est accessible sur l'url défini dans **context.xml** (paramètre **app.url**, voir [Installation Initiale](#))

Pour accéder à la version mobile il faut demander l'url **app.url/m**

La partie mobile est utilisable à partir des navigateurs utilisant WebKit (Safari pour iOS, Chrome pour Android) et par IE de Windows Phone. Si le navigateur de l'utilisateur n'est pas compatible, une redirection est proposée.