

documentation esup-connecteur-apogee

Connecteur Apogée

But

Fournir un service permet de récupérer l'identifiant dans Apogée (COD_IND) d'un étudiant connecté au portail.

Principe de fonctionnement et pré-requis

- Ce service utilise un pool de connexion tomcat pointant vers la base Apogée.
- Connaissant une requête de mapping à exécuter dans la base de données Apogée et un paramètre issu des attributs de l'utilisateur ESup à passer à cette requête, celle-ci doit nous fournir le champs COD_IND de la table INDIVIDU.
- Il fournit également un mécanisme permettant de personnaliser la récupération l'attribut de mapping à partir du login de la personne connectée. Ce mécanisme peut-être utile si on utilise une recherche de personne qui retourne comme résultat l'identifiant de la personne sélectionnée.
- Le résultat de la requête

Installation

Récupération de la distribution

- Aller sur la page de téléchargement ESup pour récupérer la dernière version du package.
- Préparation du fichier de configuration ApogeeConnector.xml (voir Configuration)
- Préparation du déploiement : modifier les chemins d'installation du fichier build.properties.
- Lancement de la commande "ant deploy", pour déployer le service dans les bons répertoires.
- Il ne s'agit pas d'un canal donc aucune action de publication n'est à entreprendre.

Configuration

Modifier le fichier ApogeeConnector.xml sachant que :

- pool : fait référence au pool tomcat de connexion vers Apogée. Attention le nom de ce pool ne doit pas comporter la partie **jdbc/** (par exemple, on fera référence au pool tomcat jdbc/Apogee par le nom Apogee)
- sqlmapping : c'est la requête à exécuter sur la base Apogée pour retourner la valeur de COD_IND. Elle comporte un et un seul paramètre.
- personattribute : attribut de l'utilisateur ESup à passer comme paramètre de la requête (doit donc être présent dans l'objet IPerson)
- classmapping : classe permettant de récupérer l'attribut de mapping à partir du login d'une personne. Elle doit implémenter l'interface IClassMapping. Lors de sa déclaration dans le fichier de configuration du connecteur, spécifier le chemin complet, c'est à dire par exemple : *org.esupportail.portal.channels.gestion.ApogeeMapping*
Le connecteur instancie alors dynamiquement la classe spécifiée et exécute la méthode déclarée dans l'interface.
Si l'on ne veut pas utiliser de classe de mapping, ne pas déclarer cette balise.
- demomapping : permet le mapping entre un login de demo et un cod_ind. Si l'on ne veut pas utiliser cette fonctionnalité, ne pas déclarer cette balise.

Cas particulier des sites qui possèdent la valeur de COD_IND dans leur LDAP

- Dans ce cas, laisser vide la valeur de sqlmapping
- Indiquer dans personattribute le nom de l'attribut LDAP qui contient la valeur de COD_IND

Utilisation

voici un exemple type d'utilisation de ce service :import org.esupportail.portal.utils.connectors.ApogeeConnector;

```
...

try {
    cod_ind = ApogeeConnector.getCOD_IND(getStaticData().getPerson());
}

catch (ConnectorException e1) {
    // enregistrement plugin fatalError
    FatalError.fatalError(getConfigActions().getActions(), "Vous n'êtes pas étudiant!");
    return null;
}
```

Cas particulier de l'utilisation de la classe *classmapping* :
Il suffit d'utiliser la méthode `getCOD_INDFromLogin()`

```
cod_ind = ApogeeConnector.getCOD_INDFromLogin(pers_id);
```