

2. Installation initiale



Prérequis : environnement technique mis en place

- 2.1 Sources
 - 2.1.1 Solution 1 : Installation rapide
 - 2.1.2 Solution 2 : Installation personnalisée
- 2.2 Paramétrage
 - 2.2.1 Paramétrage des logs
- 2.3 Paramétrage des webservices Apogée
 - A partir de la version 1.5.2
 - Versions antérieures à la 1.5.2
- 2.4 Requêtes SQL
- 2.5 Base de données
- 2.6 ElasticSearch
 - 2.6.1 Exemple
 - 2.6.2 Exemple d'installation d'ElasticSearch et de l'index apogee
 - ElasticSearch v1
 - ElasticSearch v2

2.1 Sources

2.1.1 Solution 1 : Installation rapide

La personnalisation de l'application n'est pas prévue pour cette installation (personnalisation fine du css, utilisation de beans java propres, etc..). Les étapes spécifiques à cette solution seront notées **SOLUTION 1** dans le reste du document.

- Récupérer le war de la dernière version de DossierWeb-V3 sur <https://github.com/EsupPortail/esup-mdw/releases>

2.1.2 Solution 2 : Installation personnalisée

La personnalisation de l'application est prévue pour cette installation (personnalisation fine du css, utilisation de beans java propres, etc..). Les étapes spécifiques à cette solution seront notées **SOLUTION 2** dans le reste du document.

- Récupérer le zip 'distribution' de la dernière version de DossierWeb-V3 sur <https://github.com/EsupPortail/esup-mdw/releases>
- Dézipper l'archive dans un dossier puis passer au paramétrage de l'application

2.2 Paramétrage

- Copier ou renommer le fichier **context.sample.xml** en **context.xml**. Y renseigner les paramètres ci-dessous et placer le fichier **context.xml** dans le répertoire **/conf** de Tomcat.
 - Suivant le choix de la solution dans la partie 2.1, le fichier **context.sample.xml** se trouve :
 - **SOLUTION 1** : dans **META-INF/** (peut également être placé dans le répertoire 'conf' de tomcat)
 - **SOLUTION 2** : dans **/src/main/webapp/META-INF/**

Propriété	Définition
jdbc/dbMdw	Déclaration de votre base de données MySql de monDossierWeb
jdbc/dbApogee	Déclaration de votre base de données Apogée (pour le requêtage direct sur la base) • Attention Le user doit avoir accès à la fonction APOGEE.PKB_CRD.CREDIT_ACQUIS_ELP pour permettre la récupération des crédits acquis, hors WebServices Apogée.
clefApogeeDecryptBlob	Clé Apogée pour décrypter les Blob de la base Apogée (ex : signature certificat de scolarité) Pour information, cette clé est visible dans le package PKB_CRY2
app.url	Url de l'application
startServletMobile	true pour démarrer la servlet mobile, false ne proposer que la vue desktop
cas.url	Url de votre service d'authentification CAS. Ex : https://auth.univ-ville.fr (l'application ajoute automatiquement /login)

ldap.url	Informations de connexion au ldap, respectivement :
ldap.userDn	
ldap.password	
ldap.uid.attribute	
ldap.ou.people	
ldap.ou.student	
uportal.ws	Configuration de l'authentification des enseignants via les groupes uPortal de l'ENT
uportal.groupe.s. autorises	• url des Web Services uPortal • liste des groupes autorisés (séparés par un point virgule) à accéder à MDW en tant qu'enseignant
loginApogee	Configuration de l'authentification des enseignants via la table UTILISATEUR d'Apogée
checkTesUtilisateurApogee	• Mettre loginApogee à false pour ne pas utiliser cette fonctionnalité • Mettre checkTesUtilisateurApogee à true pour vérification du témoin en service si loginApogee = true • Profil attribué aux utilisateurs Apogée : ENS (enseignant) ou GEST (gestionnaire)
profilUtilisateurApogee	
attributGroupeLdap	Configuration de l'authentification des enseignants via des groupes LDAP
listeGroupesLdap	• le nom de l'attribut contenant les groupes ldap • La liste des groupes ldap autorisant l'accès à mondossierWeb en tant qu'enseignant séparés par un point virgule. • La liste des groupes ldap autorisant l'accès à mondossierWeb en tant qu'admin séparés par un point virgule. Laisser vide pour ne pas utiliser.
listeGroupesLdapAdmin	
attributLdapEtudiant	Configuration de l'authentification des étudiants
typeEtudiantLdap	• La propriété ldap du contact désignant son type • Le type ldap désignant un étudiant (Possibilité d'indiquer plusieurs valeurs séparées par des point-virgules) • La propriété ldap du contact désignant son codetu
attributLdapCodetu	
attributLdapDocteur	Permet d'identifier les utilisateurs pour lesquels on souhaite prioritairement vérifier si l'attribution du profil "enseignant" est possible. Sans ce paramétrage, les utilisateurs cibles auraient obtenus le profil "etudiant". Typiquement le cas des doctorants (pouvant également effectuer de l'enseignement)
valeursAttributLdapDocteur	• Attribut ldap permettant de caractériser un doctorant • Une ou plusieurs valeurs (séparées par ;) de l'attribut ldap ci-dessus permettant de caractériser un doctorant
attributLdapGestionnaire	Permet d'identifier les utilisateurs auxquels attribuer le profil "gestionnaire"
valeursAttributLdapGestionnaire	• Attribut ldap permettant de caractériser un gestionnaire, ex : edupersonaffiliation • Liste des valeurs (séparées par ;) de l'attribut ldap ci-dessus permettant de caractériser un gestionnaire. Ex : staff
apoWsUsername	Indiquer de manière <u>facultative</u> un utilisateur des WebServices Apogée lors des appels aux WebServices Apogée (voir roles.xml et users.xml dans la configuration des WebServices).
apoWsPassword	
	<u>Attention</u> : Un utilisateur correctement authentifié auprès du WebService doit posséder au moins un des rôles nécessaires à l'appel des méthodes utilisées par MDW, <u>même si la méthode autorise le rôle "anonymous"</u>. Dans ce cas, le user défini dans users.xml devra posséder le rôle ROLE_ANONYMOUS.
sourceResultats	Source des résultats lors de l'appel aux Web Services Apogée pour récupérer les notes et résultats
	• Apogee ou Apogee-extraction
productionMode	Mode production. Doit être à true
debugMode	Active les traces en mode debug pour les classes java du package "mondossierweb"
enablePush	Configuration du Push :
showLoadingIndicator	
	• Activation du push. Doit être à true pour utiliser "showLoadingIndicator" • true si on doit afficher l'indicateur de loading entre certains écrans. Push obligatoire

EnablePdfSecurity	Activer la sécurité sur les pdf générés par l'application (pdf encryption) : crypte le contenu du PDF à l'aide d'un mot de passe de propriétaire généré aléatoirement pour empêcher sa modification avec Acrobat Reader.
pdf.sign.keystore.path pdf.sign.password pdf.sign.creator pdf.sign.reason pdf.sign.location pdf.sign.contact	Paramètres pour la signature électronique des pdf, respectivement : • path vers le keystore • mot de passe du keystore • "creator" à indiquer dans la signature (optionnel) • "reason" à indiquer dans la signature (optionnel) • "location" à indiquer dans la signature (optionnel) • "contact" à indiquer dans la signature (optionnel) La signature électronique par un certificat validé par une autorité de certification (le certificat de l'université) permet d'assurer l'authenticité du document. A l'ouverture du document avec Acrobat Reader il est possible de consulter le certificat et son authenticité (si validé par une autorité de certification). Bien sûr, un document signé ne peut pas être modifié. La signature est plus sécurisée que l'encryption ci-dessus.
pdf.sign.alt.tsa.url pdf.sign.alt.tsa.username pdf.sign.alt.tsa.password pdf.sign.alt.tsa.tokensize	Paramètre de la Time Stamp Authority (TSA) pour une signature compatible ALT : Nécessité de posséder un certificat délivré par une authenticité de certification externe. La signature ALT doit être activée dans l'adminView (onglet PDF) • url de la Time Stamp Authority • username auprès de la TSA • Mot de passe auprès de la TSA • Taille estimée du token
mail.smtpHost mail.from mail.to mail.startDisabling mail.stopDisabling liste.erreur.a.ignorer	Paramètres pour l'envoi des mails d'erreur, respectivement : • host smtp • champs from du mail • champs to du mail • début de la période de désactivation des mails d'erreur • fin de la période de désactivation des mails d'erreur • liste des erreurs à ignorer séparées par un point-virgule

param. elasticsearch.url param. elasticsearch.port param. elasticsearch. cluster param. elasticsearch. index param. elasticsearch. index. champrecherche param. elasticsearch. index. champcodeobjet param. elasticsearch. index. champversionobjet param. elasticsearch. index. champlibelleobjet	Configuration de la connexion a Elasticsearch • l'url du serveur • le port d'écoute • le nom du cluster • le nom de l'index contenant les objets provenant d'apogée • le nom de l'attribut du document sur lequel on effectue la recherche lors de la recherche rapide. Peut être un champ enrichi. Ex : [CODE] LIBELLE • le nom de l'attribut du document contenant le code de l'objet Apogée • le nom de l'attribut du document contenant la version de l'objet Apogée • le nom de l'attribut du document contenant le libellé de l'objet Apogée  Par convention, le type des objets doit être dans l'attribut Elasticsearch nommé "_type". Valeurs possible : CMP, VET, ETU, ELP  L'implémentation d'ElasticSearch au sein de monDossierWeb dépend de la version installée. Pour changer le version utilisée et passer en v1, il faut partir du package source et modifier le pom.xml (voir doc Personnalisation)
recherche. autocompletion.elp	Configuration de la recherche rapide • true pour autoriser l'autocomplétion sur les ELP dans la vue RechercheRapide.
param.apogee. mail.annuaire emailConverter. implementation	Configuration de la récupération du mail de l'étudiant • true pour que l'email de l'étudiant (état-civil) soit récupéré par le WS dans l'annuaire, false pour que ce soit une implémentation différente qui s'en charge. • nom de l'interface java pour la récupération du mail si param.apogee.mail.annuaire=false.
codetuFromLogin. implementation loginFromCodetu. implementation	Configuration de la récupération des identifiants Apogée d'un étudiant • Interface pour la récupération du cod_etu à partir du login • Interface pour la récupération du login à partir du codetu
etatcivil.nom. affichage	Affichage du nom des étudiants • "basique" pour utilisation du nom usuel en priorité / "standard" pour un affichage de type "nom patronymique (nom usuel)"
resultat. implementation	Configuration de la classe chargée de la récupération des notes et résultat d'un étudiant dans Apogée
serveurphoto. implementation	Configuration de la récupération des photos • Interface pour la gestion des photos
google.analytics. account piwik.tracker.url piwik.site.id	Configuration de la plateforme web analytics (facultatif). Google Analytics ou Piwik. Paramètres de Google Analytics, optionnel respectivement : • ID du compte Google Analytics Paramètres de piwik, optionnel (https://fr.piwik.org/), respectivement : ◦ L'url du piwik installé dans votre établissement ◦ L'id du projet piwik

2.2.1 Paramétrage des logs

Le fichier **logback.xml** est livré avec une configuration par défaut des logs.

Il définit notamment l'historisation des accès dans un fichier **mondossierweb-security.log** avec les informations suivantes : login, IP, profil (student ou teacher) , numéro étudiant si c'est un étudiant, accès ou non possible à l'adminView.

Il est tout à fait possible de modifier ce fichier (il faudra alors le faire à chaque installation d'une nouvelle version).

Par exemple, pour supprimer les logs d'accès, il suffit de supprimer le bloc **logger name="fr.univlorraine.mondossierweb.security"** ou de lui assigner un level **"error"**.

2.3 Paramétrage des webservices Apogée

Copier le fichier **configUrlServices.sample.properties** et renommer le en **configUrlServices.properties** puis éditer le.

Suivant le choix de la solution dans la partie 2.1, le fichier **configUrlServices.sample.properties** se trouve :

- **SOLUTION 1** : dans **/WEB-INF/classes/**
- **SOLUTION 2** : dans **/src/main/resources/**

A partir de la version 1.5.2

Renseigner :

- Les URLs des WebServices Apogée utilisés par l'application
- **OPTIONNEL** Les informations d'authentification aux WebServices Apogée si nécessaire (dépend de votre installation des WebServices Apogée)
 - Il est possible de paramétrer un user différent par service.
- **OPTIONNEL** Des headers si vous en avez l'utilité :
 - Il est possible de paramétrer des headers pour chaque service sous la forme : `api.header.NomDuService.NomDuHeader = ValeurDuHeader`
 - L'exemple ci-dessous correspond aux headers nécessaires pour utiliser l'API Manager "Gravitee".

```
# WebServices
administratifMetier.urlService=https://apo-ws.univ.fr/administratifMetier
etudiantMetier.urlService=https://apo-ws.univ.fr/etudiantMetier
pedagogiqueMetier.urlService=https://apo-ws.univ.fr/pedagogiqueMetier
geographieMetier.urlService=https://apo-ws.univ.fr/geographieMetier
offreFormationMetier.urlService=https://apo-ws.univ.fr/offreFormationMetier

#Header
api.header.administratifMetier.X-Gravitee-API-Key = xxxxxx-xxxxxx-xxxx-xxxxx
api.header.etudiantMetier.X-Gravitee-API-Key = xxxxxx-xxxxxx-xxxx-xxxxx
api.header.pedagogiqueMetier.X-Gravitee-API-Key = xxxxxx-xxxxxx-xxxx-xxxxx
api.header.geographieMetier.X-Gravitee-API-Key = xxxxxx-xxxxxx-xxxx-xxxxx
api.header.offreFormationMetier.X-Gravitee-API-Key = xxxxxx-xxxxxx-xxxx-xxxxx

#Authentification WS Apogée
etudiantMetier.apo-ws-user = userEtudiant
etudiantMetier.apo-ws-pwd = *****
```

Versions antérieures à la 1.5.2

SOLUTION 2 : Il faut également ajouter la librairie des WebServices Apogée dans le répertoire correspondant à la version utilisée (cf artifactId 'apo-webservicesclient' dans le pom.xml) sous **src/main/resources/repository/gouvleducation/apogee/apo-webservicesclient{version}** afin que maven puisse trouver la librairie lors de la compilation.

Renseigner dans **configUrlServices**, les URLs des WebServices utilisés dans l'application :

Url WebServices Apogée

```
# WebServices Dossier Etudiant
administratifMetier.urlService=https://wsapogee.univ.fr/services/AdministratifMetier
etudiantMetier.urlService=https://wsapogee.univ.fr/services/EtudiantMetier
pedagogiqueMetier.urlService=https://wsapogee.univ.fr/services/PedagogiqueMetier
geographieMetier.urlService=https://wsapogee.univ.fr/services/GeographieMetier

# Web Services Référentiel
offreFormationMetier.urlService=https://wsapogee.univ.fr/services/OffreFormationMetier

# WebServices Dossier Etudiant
administratifMetier.urlService.ssl=https://wsapogee.univ.fr/services/AdministratifMetier
etudiantMetier.urlService.ssl=https://wsapogee.univ.fr/services/EtudiantMetier
pedagogiqueMetier.urlService.ssl=https://wsapogee.univ.fr/services/PedagogiqueMetier
geographieMetier.urlService.ssl=https://wsapogee.univ.fr/services/GeographieMetier

# Web Services Référentiel
offreFormationMetier.urlService.ssl=https://wsapogee.univ.fr/services/OffreFormationMetier

#SSL
WS_SSL_MODE=false;
WS_KEY_STORE_PATH=C:/client_keystore
WS_CERTIF_STORE_PATH=C:/client_keystore
WS_KEY_PASS=password
WS_KEY_TYPE=jks
```

2.4 Requêtes SQL

Copier le fichier **apogeeRequest.sample.xml** et renommer le en **apogeeRequest.xml**.

Suivant le choix de la solution dans la partie 2.1, le fichier **apogeeRequest.sample.xml** se trouve :

- **SOLUTION 1** : dans **/WEB-INF/classes/**
- **SOLUTION 2** : dans **/src/main/resources/**

Si besoin, renseigner dans **apogeeRequest.xml** les requêtes à surcharger devant être utilisées par l'application.

En laissant le fichier tel qu'il est renseigné dans **apogeeRequest.sample.xml**, les requêtes SQL internes à l'application seront utilisées.

Sinon, plusieurs requêtes SQL sont modifiables. Dans **apogeeRequest.sample.xml** se trouve un exemple des requêtes tel qu'elles sont dans le code (java) de l'application.

Vous pouvez vous inspirer de ce fichier pour modifier ces requêtes. Les requêtes renseignées dans **apogeeRequest.xml** seront celles qui seront utilisées par l'application.



Pour plus de précisions voir le paragraphe "Requêtes SQL" dans la partie : [Personnalisation](#)

2.5 Base de données

La base de données (qui doit être préalablement créée vide) est automatiquement alimentée au démarrage de l'application.

2.6 ElasticSearch

Pour utiliser la vue Recherche Rapide de la partie enseignant de monDossierWeb, il faut disposer d'un elasticSearch. Il doit exposer un index stockant les objets Apogée que l'on souhaite proposer à la recherche rapide.



Important : La version d'ElasticSearch, la structure de l'index ainsi que la façon de l'alimenter peuvent varier tant que l'index reste paramétrable dans monDossierWeb, c'est à dire qu'il doit contenir les informations demandées dans context.xml (cf [2.2 Paramétrage](#)) et répondre aux contraintes ci-dessous

Les éléments obligatoires au bon fonctionnement de monDossierWeb

- Tous les objets doivent être dans un index unique.
- Par convention, le type de l'objet (ou document) doit se trouver dans l'attribut "`__type`". Il doit prendre les valeurs suivantes :
 - CMP quand l'objet est de type composante.
 - ELP quand l'objet est de type élément pédagogique.
 - VET quand l'objet est de type version d'étape.
 - ETU quand l'objet est de type étudiant.
- Les documents doivent également posséder :
 - un attribut correspondant au libellé
 - un attribut correspondant au code
 - un attribut correspondant à la version
 - un attribut qui va servir pour la recherche rapide :
 - Il peut s'agir simplement de l'attribut libellé ou encore d'un attribut concaténant le code et le libellé de l'objet. Pour exemple, à l'Université de Lorraine nous avons un attribut formaté comme suit : `[CODE] LIBELLE`

2.6.1 Exemple

Voici un exemple de l'index utilisé à l'Université de Lorraine :

_index	_type	_id	_score ▼	COD_OBJ	COD_VRS_OBJ	LIB_DESC_OBJ	LIB_OBJ
apogee	CMP	CMPE010	1	E01	0	[E01] PREMIERE ANNEE COMMUNE ETUDES DE SANTE	PREMIERE ANNEE COMMUNE ETUDES DE SANTE
apogee	CMP	CMPPFAJ0	1	FAJ	0	[FAJ] UFR STMIA site Bar le Duc	UFR STMIA site Bar le Duc
apogee	CMP	CMPPFC00	1	FC0	0	[FC0] UFR SCIENCES FONDAMENTALES ET APPLIQUEES	UFR SCIENCES FONDAMENTALES ET APPLIQUEES
apogee	CMP	CMPPGE00	1	GE0	0	[GE0] UFR SCIENCES DU LANGAGE	UFR SCIENCES DU LANGAGE
apogee	CMP	CMPPH00	1	H00	0	[H00] IUT CHARLEMAGNE NANCY	IUT CHARLEMAGNE NANCY
apogee	CMP	CMPPHFB0	1	HFB	0	[HFB] IUT Antenne Saint-Avoid	IUT Antenne Saint-Avoid
apogee	CMP	CMPPHGO0	1	HGO	0	[HGO] IUT SAINT DIE DES VOSGES	IUT SAINT DIE DES VOSGES
apogee	ELP	ELP+CML41510	1	+CML4151	0	[+CML4151] CM Mise en tourisme du territoire UE 405B L2 LLCE	CM Mise en tourisme du territoire UE 405B L2 LLCE
apogee	ELP	ELP+CMM11130	1	+CMM1113	0	[+CMM1113] CM Linguistique et sémantique 1 UE 701A	CM Linguistique et sémantique 1 UE 701A
apogee	ELP	ELP+CMM31330	1	+CMM3133	0	[+CMM3133] CM Aires culturelles 3 UE903R M2 LCE	CM Aires culturelles 3 UE903R M2 LCE
apogee	ELP	ELP+KXLVN200	1	+KXLVN20	0	[+KXLVN20] CHOIX LANGUES VIVANTES - Semestre B	CHOIX LANGUES VIVANTES - Semestre B
apogee	ELP	ELP+LBL#0100	1	+LBL#010	0	[+LBL#010] UE LIBRE	UE LIBRE
apogee	ELP	ELP+LCM41510	1	+LCM4151	0	[+LCM4151] TD - Histoire des 20ème et 21ème siècles	TD - Histoire des 20ème et 21ème siècles
apogee	ELP	ELP+LLB41010	1	+LLB4101	0	[+LLB4101] UE LIBRE PARCOURS TRADUCTION S4	UE LIBRE PARCOURS TRADUCTION S4
apogee	VET	VET0KTNB2300	1	0KTNB2	300	[0KTNB2/300] DESMBD 2ème année	DESMBD 2ème année
apogee	VET	VET0KTNC4300	1	0KTNC4	300	[0KTNC4/300] DESCO 4ème année	DESCO 4ème année
apogee	VET	VET0KTNO3300	1	0KTNO3	300	[0KTNO3/300] DESODF 2ème année	DESODF 2ème année
apogee	VET	VET11X304700	1	11X304	700	[11X304/700] DUT INFORMATIQUE DECALE 3A (semestre 4)	DUT INFORMATIQUE DECALE 3A (semestre 4)
apogee	VET	VET1IAG41411	1	1IAG41	411	[1IAG41/411] Ingénieur Ensaia AGR 2A	Ingénieur Ensaia AGR 2A

2.6.2 Exemple d'installation d'ElasticSearch et de l'index apogee



Rappel : La version d'ElasticSearch, peut varier tant que l'index reste paramétrable dans monDossierWeb (context.xml (cf [2.2 Paramétrage](#))).

Il est tout à fait possible de monter de version ou d'avoir un workflow de renseignement de l'index totalement différent tant que l'index contient les informations nécessaires à monDossierWeb.

Ce paragraphe n'est qu'un exemple pour aider les universités qui débuteraient avec ElasticSearch.

Important : L'implémentation d'ElasticSearch au sein de monDossierWeb dépend de la version installée. Pour changer le version utilisée et passer en v1, il faut partir du package source et modifier le pom.xml (voir [doc Personnalisation](#))

ElasticSearch v1

Pour construire cet index d'exemple nous utilisons une "river", qui permet de créer un index à partir d'une requête SQL.



Attention : le plugin 'river' n'étant plus disponible depuis la v2.0 d'ElasticSearch, vous devez installer une version antérieure pour utiliser l'exemple ci-dessous.

NB : Le fonctionnement a été également validé avec elasticSearch en version 1.7.0.3 et le plugin river en version 1.5.0.5

1) Télécharger et dézipper Elasticsearch ([elasticsearch-1.3.4.official.zip](#))

Dans config/elasticsearch.yml décommenter et renseigner les paramètre **cluster.name** et **node.name**. Le cluster est à renseigner dans [context.xml](#)

2) démarrer Elasticsearch en lançant bin/elasticsearch depuis le répertoire d'installation,

3) Installer le plugin head :

```
./bin/plugin -install mobz/elasticsearch-head
```

Vous pouvez voir si le plugin est fonctionnel en vous rendant sur : http://localhost:9200/_plugin/head/ via votre navigateur web

4) Installer le plugin river-jdbc pour Elasticsearch . Il suffit de lancer la commande suivante :

```
./bin/plugin --install river-jdbc --url http://xbib.org/repository/org/xbib/elasticsearch/plugin/elasticsearch-river-jdbc/1.3.4.7/elasticsearch-river-jdbc-1.3.4.7-plugin.zip
```

Au cas où, voici l'archive du plugin : [elasticsearch-river-jdbc-1.3.4.7-plugin.zip](#)

Pour lister les plugins installés :

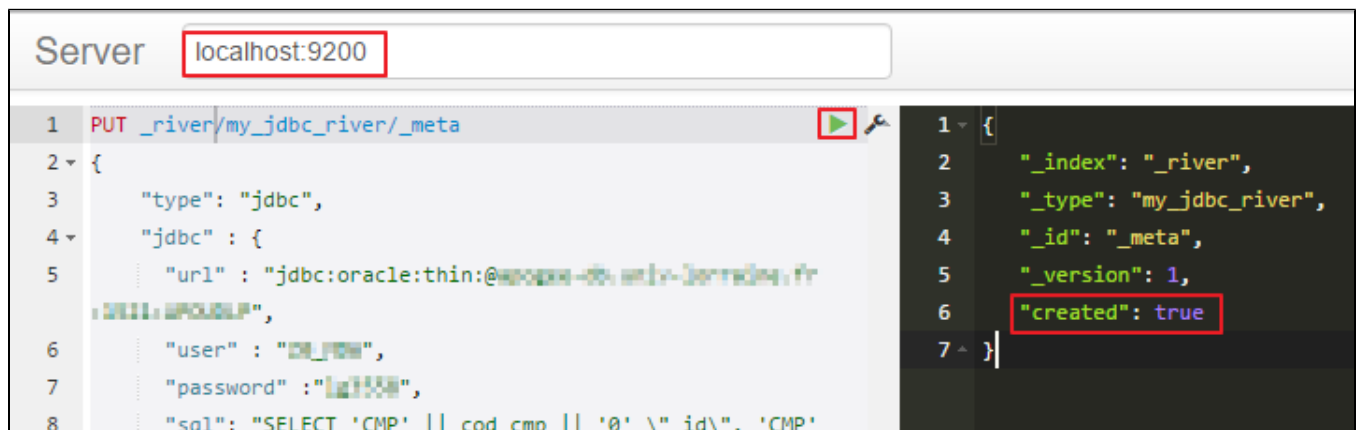
```
./bin/plugin --list
```

5) Télécharger le driver oracle jdbc.

6) Copier le jar dans elasticsearch/plugins/river-jdbc

7) Redémarrer ElasticSearch

8) Créer la river avec le plugin Sense pour Chrome (le nom donné à l'index sera à renseigner dans [context.xml](#)):



Exemple de création de river permettant de récupérer les composantes, VET, ELP, et étudiants valides sur n et n-1:

```
PUT _river/my_jdbc_river/_meta
{
  "type": "jdbc",
  "jdbc": {
    "url": "jdbc:oracle:thin:@apogee.univ.fr:1521:APOGEE",
    "user": "userApogee",
    "password": "userPwd",
    "sql": "SELECT 'CMP' || cod_cmp || '0' \"_id\", 'CMP' \"_type\", cod_cmp COD_OBJ, 0 COD_VRS_OBJ, '[' || cod_cmp || ']' || lib_cmp LIB_DESC_OBJ, lib_cmp LIB_OBJ FROM composante UNION SELECT DISTINCT 'VDI' || vdi.cod_dip || vdi.cod_vrs_vdi \"_id\", 'VDI' \"_type\", vdi.cod_dip COD_OBJ, vdi.cod_vrs_vdi COD_VRS_OBJ, '[' || vdi.cod_dip || '/' || vdi.cod_vrs_vdi || ']' || vdi.lib_web_vdi LIB_DESC_OBJ, vdi.lib_web_vdi LIB_OBJ FROM version_diplome vdi, diplome d, VDI_FRACTIONNER_VET vfv WHERE vdi.cod_dip = d.cod_dip and vfv.COD_DIP=d.cod_dip and vfv.COD_VRS_VDI=vdi.COD_VRS_VDI and vfv.DAA_FIN_RCT_VET > (select cod_anu - 2 from annee_uni where eta_anu_iae = '0') UNION SELECT DISTINCT 'VET' || vet.cod_etp || vet.cod_vrs_vet \"_id\", 'VET' \"_type\", vet.cod_etp COD_OBJ, vet.cod_vrs_vet COD_VRS_OBJ, '[' || vet.cod_etp || '/' || vet.cod_vrs_vet || ']' || vet.lib_web_vet LIB_DESC_OBJ, vet.lib_web_vet LIB_OBJ FROM version_etape vet, etape e, VDI_FRACTIONNER_VET vfv WHERE vet.cod_etp = e.cod_etp and vfv.COD_ETP=e.cod_etp and vfv.COD_VRS_VET=vet.COD_VRS_VET and vfv.DAA_FIN_RCT_VET > (select cod_anu - 2 from annee_uni where eta_anu_iae = '0') UNION SELECT 'ELP' || cod_elp || '0' \"_id\", 'ELP' \"_type\", cod_elp COD_OBJ, 0 COD_VRS_OBJ, '[' || cod_elp || ']' || lib_elp LIB_DESC_OBJ, lib_elp LIB_OBJ FROM element_pedagogi e UNION select 'ETU' || IND.COD_ETU || ' ' || IND.LIB_PR1_IND || ' ' || NVL(IND.LIB_NOM_USU_IND, IND.LIB_NOM_PAT_IND) LIB_DESC_OBJ, IND.LIB_PR1_IND || ' ' || NVL(IND.LIB_NOM_USU_IND, IND.LIB_NOM_PAT_IND) LIB_OBJ from individu ind, INS_ADM_ANU iaa where IAA.COD_IND=ind.COD_IND and IND.COD_ETU is not null and IAA.COD_ANU > (select cod_anu - 2 from annee_uni where eta_anu_iae = '0')",
    "index": "apogee"
  }
}
```

A noter que les "river" peuvent être automatisés pour mettre à jour l'index régulièrement. Exemple :

```
PUT _river/my_jdbc_river/_meta
{
  "type": "jdbc",
  "jdbc": {
    "url" : "jdbc:oracle:thin:@apogee.univ.fr:1521:APOGEE",
    "user" : "userApogee",
    "password" : "userPwd",
    "schedule": "0 0/15 5-23 * * ?",
    "sql": "REQUETE SQL SUR MESURE",
    "index" : "apogee"
  }
}
```

On peut voir sur http://localhost:9200/_plugin/head/ que l'index est créé et renseigné

The screenshot shows the ElasticSearch v2 web interface. At the top, there's a navigation bar with 'Elasticsearch' and a search bar containing 'http://localhost:9200/'. Below this, there are tabs for 'Aperçu', 'Index', 'Navigateur', 'Recherche Structurée [+]', and 'Autres requêtes [+]'.

The main section is titled 'Aperçu du cluster'. It includes a 'Sort Cluster' dropdown, a 'View Aliases' dropdown, and an 'Index Filter' input field. Below this, there's a table-like structure showing two indices:

- _river**: size: 10.1ki (10.1ki), docs: 2 (2). It has 'Info' and 'Actions' buttons.
- apogee**: size: 42.1Mi (42.1Mi), docs: 281 849 (281 849). It has 'Info' and 'Actions' buttons.

Below the index information, there are two sections: 'Unassigned' and 'Corsair'. Each section has a grid of node status indicators (0-4). The 'apogee' index is highlighted with a red box.

ElasticSearch v2

Pour construire cet index d'exemple nous utilisons logstash, qui permet de créer un index à partir d'une requête SQL.

1) Télécharger et dézipper Elasticsearch ([elasticsearch-2.3.4.zip](#))

Dans config/elasticsearch.yml décommenter et renseigner les paramètres **cluster.name** et **node.name**. Le cluster est à renseigner dans **context.xml**

2) démarrer Elasticsearch en lançant bin/elasticsearch depuis le répertoire d'installation,

3) Installer le plugin head :

```
./bin/plugin -install mobz/elasticsearch-head
```

Vous pouvez voir si le plugin est fonctionnel en vous rendant sur : http://localhost:9200/_plugin/head/ via votre navigateur web.

Pour lister les plugins installés :

```
./bin/plugin list
```

4) Télécharger Logstash ([logstash-2.3.4.zip](#)) et le dézipper aux côtés d'ElasticSearch.

5) Télécharger le driver oracle jdbc.

6) Copier le jar dans le répertoire lib de logstash

7) Depuis le répertoire bin de logstash, lancer la commande suivante pour installer le plugin "input-jdbc" :

```
./plugin install logstash-input-jdbc
```

8) Dans le répertoire qui contient Elasticsearch et Logstash, créer un fichier **apogee-test.conf** à partir du fichier exemple suivant permettant de récupérer les composantes, VET, ELP, et étudiants valides sur n et n-1 : [apogee-test-exemple.conf](#)

En y configurant :

- l'accès à Apogée,
- le driver utilisé
- le nom de l'index (le nom donné à l'index sera à renseigner dans [context.xml](#) de MonDossierWeb).
- l'url d'accès à l'ElasticSearch

A noter que le plugin peut être automatisé pour mettre à jour l'index régulièrement. Pour cela il suffit d'ajouter un scheduler. Exemple :

schedule une fois par heure entre 5h et 23h

```
...
jdbc_driver_class => "Java::oracle.jdbc.driver.OracleDriver"
schedule => "0 5-23 * * *"
statement => "SELECT ..."
```

9) Puis, depuis le répertoire bin, de logstash :

```
./logstash -f ../../apogee-test.conf
```

i Attention : Il est nécessaire de relancer la commande ci-dessus à chaque redémarrage d'ElasticSearch.
L'idéal est de l'ajouter dans un script de démarrage juste après la commande de démarrage d'ES .

On peut voir sur http://localhost:9200/_plugin/head/ que l'index est créé et renseigné

The screenshot shows the ElasticSearch web interface at <http://localhost:9200/>. The top navigation bar includes 'Aperçu', 'Index', 'Navigateur', 'Recherche Structurée [+]', and 'Autres requêtes [+]'. The 'Aperçu du cluster' section is active, displaying the 'apogee' index. The index details show a size of 55.2Mi and 265,752 documents. Below the index name, there are buttons for 'Info' and 'Actions'. The 'Unassigned' section shows five nodes (0-4) in a grey state. The 'noeud-0' section shows five nodes (0-4) in a green state, indicating they are healthy and assigned to the index.