

Master 2 MIASHS WIC Alternant

Blarre Pierre

PicavetNicolas

David Jérôme  
Manuel Atencia

## Application Photo « My Lil'Album »

JEE + Web Semantic

# Sommaire

Introduction.....	3
Contenu de l'archive.....	3
Guide de déploiement de l'application.....	3
Si vous ne souhaitez pas importer de données.....	3
Si vous souhaitez importer les données.....	4
Exemple de requêtes.....	4
Ontologie (retrouvez cette image dans l'archive du projet : ontology.png).....	6
Conclusion.....	7

# Introduction

Le projet et ses sources sont disponibles ici :

<https://github.com/NicolasPicavet/BlarrePicavetSempic>

## Contenu de l'archive

- **BlarrePicavetSempic.zip** : Le projet Netbean de l'application
- **images.zip** : les fichiers images du projet
- **sempic\_data.nq.gz** : les données à importer dans Fuseki
- **data.sql** : les données à insérer dans la base de données JavaDB (L'archive BlarrePicavetSempic.zip contient aussi un repertoire « newsempic » qui est la DB Java, il est possible de la lier directement)
- **sempic.ttl** Le fichier de configuration Fuseki

## Guide de déploiement de l'application

### Si vous ne souhaitez pas importer de données

- Importer l'archive du projet dans Netbeans
- Dans Netbeans, ouvrir le fichier WebPages → WEB-INF → web.xml, puis changer les valeurs des paramètres : upload.location, imgFolder et thumbFolder avec le chemin du repertoire dans lequel vous avez exporté les images.
- Placer sempic.ttl dans le repertoire « configuration » de Fuseki
- Lancer Fuseki
- De retour dans Netbeans :
- Ouvrir le fichier persistence.xml and modifier la Validation Strategy à « Drop & Create »
- lancer un « Clean + Build » sur le projet puis une runL'application devrait démarrer et une page s'ouvrir dans votre navigateur.
- Si le navigateur ne s'ouvre pas, aller à l'url : <http://localhost:8080/BlarrePicavetSempic/>
- Si vous obtenez une 404, vérifier que Glassfish et Fuseki sont bien en marche
- Si vous obtenez une autre erreur, vérifier que la DB a bien été créée.

## Si vous souhaitez importer les données

- Exporter l'archive « images.zip » dans un répertoire de votre choix
- Importer l'archive du projet dans Netbeans
- Dans Netbeans, ouvrir le fichier WebPages → WEB-INF → web.xml, puis changer les valeurs des paramètres : upload.location, imgFolder et thumbFolder avec le chemin du répertoire dans lequel vous avez exporté les images.
- Placer **sempic.ttl** dans le repertoire « configuration » de Fuseki
- Lancer Fuseki, aller sur l'administration des datasets, (<http://localhost:3030/manage.html>) puis importer le fichier sempic\_data.nq.gz
- De retour dans Netbeans, lancer un « Clean + Build » sur le projet
- Aller dans l'onglet « Services » et vérifier que la base de données « newsempic » a bien été créée.
- Faire un click droit sur le lien « jdbc:derby://localhost..... » et sélectionner « Execute Command ». Copier coller le contenu SQL du fichier « data.sql » puis lancer l'exécution
- Lancer un « Run » sur le projet
- L'application devrait démarrer et une page s'ouvrir dans votre navigateur.
- Si le navigateur ne s'ouvre pas, aller à l'url : <http://localhost:8080/BlarrePicavetSempic/>
- Si vous obtenez une 404, vérifier que Glassfish et Fuseki sont bien en marche
- Si vous obtenez une autre erreur, vérifier que la DB a bien été créée.

Vous pouvez vous créer un compte utilisateur ou bien utiliser le compte administrateur suivant :

**email** : [admin@admin.com](mailto:admin@admin.com)

**mdp** : admin

## Exemple de requêtes

### Récupérer les classes et leur label

```
CONSTRUCT { ?s <http://www.w3.org/2000/01/rdf-schema#label> ?o } WHERE { ?s  
<http://www.w3.org/2000/01/rdf-schema#subClassOf> <http://miashs.univ-grenoble-  
alpes.fr/ontologies/sempic.owl#Animal> . ?s <http://www.w3.org/2000/01/rdf-schema#label> ?o . }  
ORDER BY ASC(?o)
```

```
CONSTRUCT { ?s <http://www.w3.org/2000/01/rdf-schema#label> ?o } WHERE { ?s  
<http://www.w3.org/2000/01/rdf-schema#subClassOf> <http://miashs.univ-grenoble-  
alpes.fr/ontologies/sempic.owl#Person> . ?s <http://www.w3.org/2000/01/rdf-schema#label> ?o . }  
ORDER BY ASC(?o)
```

### Recherche Aude ou Medor pour « Who's in the picture ? »

```
SELECT ?p WHERE { ?p <http://miashs.univ-grenoble-  
alpes.fr/ontologies/sempic.owl#inThePicture> ?o. FILTER ( ?o = <http://miashs.univ-grenoble-  
alpes.fr/ontologies/sempic.owl#Aude> || ?o = <http://miashs.univ-grenoble-  
alpes.fr/ontologies/sempic.owl#Medor> ) }
```

### **Recherche Animal pour « What's in the picture »**

```
SELECT ?p WHERE {?p <http://miashs.univ-grenoble-alpes.fr/ontologies/sempic.owl#depicts> ?o.  
FILTER ( ?o = <http://miashs.univ-grenoble-alpes.fr/ontologies/sempic.owl#Animal> )}
```

Les résultats des requêtes (ID des photos) sont ensuite compilés et une requête SQL est effectuée pour récupérer les photos.

# Ontologie (retrouvez cette image dans l'archive du projet : ontology.png)



# Conclusion

Nous avons réussi à construire une application JEE utilisant apache-jena pour interagir avec une ontologie OWL.

## **Nous avons pu créer :**

- Un CRUD (Create Read Update Delete) utilisateur et l'authentification
- Un CRUD Album
- Un CRUD Picture, ainsi qu'un bean pour « Servir » les images
- Le controller Picture a ensuite été amélioré pour gérer les données RDF associées aux images
- Une action « search » qui permet de faire une recherche sur les données RDF et faire remonter les images associées

## **Nous n'avons pas pu ou pas eu le temps de créer :**

- La gestion de « When was the picture taken ». Une date est stockée mais la recherche associée au temps ne fonctionne pas.
- L'édition des images n'est pas complète.
- La recherche reste basique, nous n'avons pas eu le temps de compléter les requêtes complexes du type : « All pictures of Manuel Atencia's friends. ».

Avec plus de temps nous aurions pu finaliser la recherche complexe et perfectionner la partie JEE, cependant ce projet nous a apporté de nombreuses connaissances sur les différentes technologies employées et sur l'utilisation de la sémantique web dans une application.