



StrikeRepo

(Développeur: Shineeeeeeh)



Présentation du projet

Ce projet a été commencé à cause du fait que suite au passage de java à la version 11 etc... Notre ancien gestionnaire de repo pour Maven, ne marche plus !

Du coup, j'ai commencé à créer un gestionnaire de repo pour Maven en utilisant la librairie "ExpressJS" en NodeJS.

Le but est de pouvoir aussi faire ces choses suivantes:

- Création d'identifiant unique
- Chaque identifiant à accès à un repo différents
- Créer facilement des repo
- Ajouter facilement des artefacts
- Pouvoir gérer facilement les utilisateurs
- Création simplifier d'un repo
- Uniquement pour maven



Utilisateurs

Les utilisateurs sont stocker dans le dossier `"data/users"`.

Ils sont stocker dans un fichier **JSON** qui ont comme nom:

`${pseudo}.json` | `${pseudo}` est le pseudo de l'utilisateur !

Le fichier JSON à ce modèle suivant :

Username (String) Pseudonyme
Password (String) Mot de passe encodé en MD5
UUID (String) Identifiant unique pour l'utilisateur
isAdmin (Boolean) A-t-il les privilèges admin ? Défaut: false

Ce qui nous donne par exemple ceci avec le fichier `admin.json`:

```
{
  "username" : "admin",
  "password" : "202cb962ac59075b964b07152d234b70",
  "uuid" : "d1577ae0-37fb-426a-8a97-8d7d5e71c796",
  "isAdmin" : true
}
```

(PS: le mot de passe du compte "admin" est "123")



Repository

Les repository permettra de stocker les artefacts sur Maven.

Ils seront stocker dans le dossier "data/repositories", sous un fichier JSON ce nommant :

`${nom du repo}.json` | `${nom du repo}` est le nom du repo

Ce fichier contiendra ce modèle suivant :

Name (String) : Nom du repo

Access (String Table) : Liste des utilisateurs qui y ont accès

onlyAdmin (Boolean) : Si seulement réserver au admins

Ce qui nous donne par exemple avec le fichier exemple.json :

```
{
  "name": "exemple",
  "access" : [
    "shine"
  ],
  "onlyAdmin": false
}
```

Les artefacts seront stocker dans un dossier du même nom que le repo dans le dossiers "data/repositories/".



Artefacts

Les artefacts sont des fichiers qu'on initialise dans le "pom.xml" pour les intégrer dans le projet Java sous Maven.

Routes utiliser par Maven pour les artefacts :

Schéma:

`${lien_repo}` | Lien du repo

`${group_id}` | ID du groupe

`${name}` | Nom de l'artefact

`${version}` | Version

Lien principal aka `${lien_artefact}`:

`"${lien_repo}/${group_id}/${name}/${version}"`

Pom: `${lien_artefact}/${name}-${version}.pom`

Jar: `${lien_artefact}/${name}-${version}.jar`



Connexion

Modèle:

Connexion

Pseudo

Se connecter

Le bouton "Se connecter" redirige sur une requête en POST de la page "/login".



Navbars

Accueil

Repos

Admin

Deconnexion

Navbar de base, Admin apparaît uniquement quand tu es admin !

Navbar panel Admin
(Uniquement vu par les admins)



Informations supplémentaires

Ce projet sera sous NodeJS avec la librairie "ExpressJS".

Avec l'utilisation du template engine "Pug".

L'utilisation de la framework CSS "Semantic CSS".

Express

