

JSON

Json-lib

Utiliser Json-lib

Pour cette démonstration, nous allons introduire [Json-lib](#). D'autres bibliothèques pour JSON sont disponibles depuis la page officielle du format JSON. Pour vos travaux, vous devez utiliser seulement Json-lib.

Sur Moodle, au niveau du cours 2, vous pouvez trouver un fichier archive .zip regroupant toutes les dépendances (plusieurs Fichiers .jar) nécessaires à Json-lib pour fonctionner.

Voici la procédure pour ajouter Json-lib à votre projet Java en utilisant l'archive disponible sur Moodle :

1. Commencer par télécharger l'archive disponible sur Moodle contenant tous les fichiers jar.
2. Créer un dossier dans votre projet nommé « lib », pour ce faire :
 - a) Clic droit sur le projet
 - b) Sélectionner `New` après `Other...`
 - c) Puis sélectionner la catégorie `Other` après `Folder`.
3. Copier et coller le contenu de l'archive (les fichiers .jar) dans le dossier « lib ».
4. Importer les bibliothèques dans votre projet Java
 - a) Clic droit sur le projet
 - b) Sélectionner `Properties` après `Libraries`.
 - c) Puis sélectionner `Add JAR/Folder`
 - d) Spécifier ensuite le répertoire en question.
5. Json-lib est maintenant intégré dans votre projet Java.

N'hésitez pas à consulter l'[API de Json-lib](#) pour plus de détails.

Exemple 1

Dans ce premier exemple, nous allons parser une chaîne JSON à l'aide de Json-lib.

1. On commence par charger depuis votre programme java le fichier JSON disponible sur Moodle (catalogue.json)
2. On doit ensuite convertir ce dernier en `String`.

Vous pouvez utiliser le Jar File Manager fourni sur Moodle.

3. Étant donné que c'est un tableau JSON (et non un objet), on doit donc récupérer la liste des livres.

Vous pouvez utiliser la méthode [`JSONArray.fromObject\(Object obj\)`](#).

4. On doit ensuite afficher le nombre de livres dans la liste.

Vous pouvez utiliser la méthode [`JSONArray.size\(\)`](#).

La console doit ainsi afficher le message suivant :

```
« Il y a 5 livre(s) dans le catalogue. »
```

Exemple 2

Dans ce deuxième exemple, on nous demande de lister les titres des livres du catalogue uniquement pour les livres parus après 2009.

La console doit ainsi afficher le message suivant :

```
Livres parus après 2009:  
* Systèmes de gestion de bases de données par l'exemple  
* Fundamentals of Database Systems  
Il y a 5 livre(s) dans le catalogue, dont 2 paru(s) après 2009.
```

1. On doit faire une itération sur un tableau JSON pour vérifier notre condition.

Vous pouvez utiliser la méthode [`JSONArray.getJSONObject\(int index\)`](#).

Exemple 3

Dans ce troisième exemple, on nous demande de lister les prix des livres encore disponibles.

La console doit ainsi afficher le message suivant :

Prix des livres disponibles:

- * Agile Software Development, Principles, Patterns, and Practices: 29.9\$*
- * Systèmes de gestion de bases de données par l'exemple: 110.2\$*
- * Fundamentals of Database Systems: 19.9\$*

Il y a 3 livre(s) disponible(s).

1. On doit faire une itération sur un tableau JSON pour vérifier notre condition.

Vous pouvez utiliser la méthode [JSONArray.getJSONObject\(int index\)](#)

Exemple 4

Dans ce quatrième exemple, on nous demande de produire le document JSON suivant :

```
{
  "id": "1321033823",
  "total": 49.9,
  "date": "11/11/2011",
  "validation": true,
  "livres": [{
    "id": "1",
    "titre": "Database System Concepts"
  }]
}
```

1. On doit produire le document JSON selon le modèle fourni.

Vous pouvez utiliser la méthode [JSONObject.accumulate\(\)](#)

Vous pouvez utiliser [JSONLint](#) pour vérifier que votre JSON est valide.

Exemple 5

Dans ce cinquième exemple, on nous demande de créer une commande en format JSON à partir de notre catalogue de livres JSON.

1. On doit parcourir le catalogue tout en ajoutant à la commande tous les livres dont le prix est inférieur à 100\$.
2. On doit ensuite calculer le prix total de la commande en conséquence et l'intégrer.

Vous pouvez utiliser la méthode [JSONObject.accumulate\(\)](#)

Vous pouvez utiliser [JSONLint](#) pour vérifier que votre JSON est valide.

3. On doit finalement enregistrer le JSON dans un fichier order.json.

Vous pouvez utiliser le Jar File Manager fourni sur Moodle.

On nous demande donc de produire le document JSON suivant :

```
{
  "id": "1321033823",
  "total": "89,70",
  "date": "13/01/2016",
  "validation": true,
  "livres": [{
    "id": "1",
    "titre": "Coder proprement",
    "auteur": "Robert Martin",
    "disponible": false,
    "prix": 39.9,
    "annee": 2009
  }, {
    "id": "2",
    "titre": "Agile Software Development, Principles, Patterns,
and Practices",
    "auteur": "Robert Martin",
    "disponible": true,
    "prix": 29.9,
    "annee": 2002
  }, {
    "id": "4",
    "titre": "Fundamentals of Database Systems",
    "auteur": "R. Elmasri",
    "disponible": true,
    "prix": 19.9,
    "annee": 2011
  }]
}
```

La [Correction](#) est disponible sur Git (intentionnellement, nous avons inclus tout le projet y compris les librairies pour en faciliter le clone).