

## Javascript : exercices séance 7

### Partie HTTPServlet

Nous allons mettre sur pied un site Web qui calcule des expressions mathématiques.

L'architecture sera la suivante :

1. Créez une page HTML contenant un champ textarea dans lequel l'utilisateur rentre sa formule.
2. Ajoutez un bouton pour envoyer la requête. Cette dernière aura la forme d'un GET envoyé au serveur et fournissant la formule du textarea dans le paramètre expr. En d'autres termes, il faudra rediriger la page vers quelque chose ressemblant à `http://127.0.0.1/Calc?expr=2+2`  
Comment redirige-t-on encore en Javascript ?  
Ce que l'utilisateur entre dans le textarea finit par se retrouver dans l'URL. Il pourrait donc en profiter pour abuser le comportement de cette URL en y mettant des caractères non prévus, par exemple &. Tout paramètre envoyé par un GET doit donc être encodé. Consultez [https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets\\_globaux/encodeURIComponent](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/encodeURIComponent) pour la solution standard à ce problème.
3. Créez un serveur en Java :
  - a. Créez un répertoire lib à la racine de votre projet et décompressez-y le contenu de Jetty.zip qui se trouve sur l'ecampus. Faites refresh (F5) sur la racine de votre projet, sélectionnez les fichiers .jar du répertoire lib et click-droit/Build Path/Add To Build Path.
  - b. Créez une classe et dans sa méthode main, copiez les lignes de code de la présentation permettant d'utiliser Jetty en mode embedded.
  - c. Servez les fichiers requis pour le front-end à l'aide d'un DefaultServlet.
  - d. Créez une servlet pour recevoir les GET de calcul d'expressions mathématiques.
  - e. Pour le calcul à proprement dit, vous pouvez utiliser la librairie exp4j : <http://projects.congrace.de/exp4j/index.html> .
  - f. La réponse renvoyée doit être simplement un document HTML dont le body est la réponse.
  - g. Configurez votre serveur pour que tout fonctionne !
4. Bonus : améliorez le rendu du résultat :
  - a. Rappelez la formule évaluée (afficher `1+1=2.0` plutôt que se contenter d'afficher juste `2.0`).
  - b. Rendez l'affichage plus joli.
  - c. En plus d'afficher le résultat, permettez d'entrer une nouvelle formule mathématique.
  - d. Suivez votre imagination...

### Partie Genson

Lisez attentivement <http://owlike.github.io/genson/GettingStarted/>

## Ex 1.

Dans un fichier calc.json, entrez ceci :

```
{"x":4, "y":10, "eval":"x+y"}
```

L'idée de ce JSON est de décrire une expression mathématique avec des variables. La clef « eval » contient toujours l'expression à évaluer. Les autres clefs sont par contre libres, chacune représentant une variable de l'expression mathématique.

Ecrivez un programme java qui lit le fichier calc dans une chaîne de caractères.

```
String content = new  
String(Files.readAllBytes(Paths.get("calc.json")));
```

Utilisez genson et exp4j pour créer l'expression, placer les valeurs des variables et afficher le résultat.

Bonus : changez l'application web de votre point précédent pour qu'elle aussi fonctionne sur base d'un tel JSON.

## Ex 2.

Une machine de l'IPL est définie par :

- Son local = un numéro de local
- Sa localisation dans ce local = une chaîne de caractère
- Un modèle de CPU
- Une fréquence de CPU
- Une taille de mémoire en GB
- Une taille de disque dur en GB
- Un modèle de carte graphique
- Une date d'achat

Décidez d'une classe Java et d'un JSON qui décrivent cette machine. Faites attention au format de data : dans le JSON il faut utiliser une chaîne au format JJ/MM/AAAA et en Java utiliser une classe Date.

Ecrivez trois fichiers JSON : un pour votre machine, un pour la machine à votre gauche (ou derrière ou devant), un pour la machine à votre droite (ou derrière ou devant).

Ecrivez une classe Java, modèle d'une machine, ainsi que sa méthode toString().

Ecrivez une application de test qui lit les fichiers json, les transforme en objets Java en utilisant vos classes, affiche le contenu avec le toString(), retransforme la classe en chaîne de caractère au format JSON proprement formaté.