

TP n°4

Toutes les ressources seront déclarées dans un package fr.tp.isima

Vous pouvez partir de vos propres sources ou de la correction. Toutefois la correction possède quelques éléments facilitant votre travail sur cette nouvelle version.

Exercice : Display Quote v2.0 !

Les utilisateurs apprécient de pouvoir naviguer de citation en citation mais trouvent le rafraîchissement désagréable.

Adaptez display-quote 1.0 pour pouvoir avoir une navigation en Ajax plus sympathique. Les échanges entre le client et le serveur auront lieu au format « json ».

Créez un service en JSON.

Pour pouvoir réaliser la navigation en AJAX, vous pouvez procéder en réalisant un service JSON invoqué en appelant une Servlet.

En résumé, une Servlet va devoir rendre du Json contenant la navigation. Appelons la « loadQuote ».

a) Créez d'abord un service simple rendant juste le citation

Voici un exemple du flux JSON généré.

```
{ "author": "Georges Clemenceau", "content": "Un traître est un homme politique qui quitte son parti pour s'inscrire à un autre. Par contre, un converti est un homme politique qui quitte son parti pour s'inscrire au votre." }
```

Utilisez l'API JsonObjectBuilder vu en cours.

b) Intégrez le en Ajax

Intégrez dans votre page le rafraîchissement de la requête en Ajax. Vous pouvez simplement afficher toujours le même résultat suite au clic sur les boutons – l'équivalent du bouton de rafraîchissement du TP3. Cela vous permettra de valider la communication entre votre serveur et votre client.

c) Intégrez la navigation

Intégrez la navigation pour cela modifiez votre flux json (donc votre serializer pour générer un flux contenant les informations de navigation) :

- l'id de la direction ;
- le prochain numéro de quote accessible si l'on clique sur le bouton correspondant à cette navigation ;
- s'il est possible de cliquer sur le bouton à partir de la citation actuelle

Voici un exemple du flux JSON que vous pouvez utiliser (ici le quoteNumber est à 1) :

```
{ "author": "xxx", "content": "xxx",  
  "directions": [  
    { "id": "first", "targetIndex": 1, "enabled": false },  
    { "id": "previous", "targetIndex": 0, "enabled": false },  
    { "id": "next", "targetIndex": 2, "enabled": true },  
    { "id": "last", "targetIndex": 8, "enabled": true } ] }
```

Les directions sont envoyées sous forme de tableau. Les tableaux peuvent être construits à partir des `JsonArrayBuilder` et ajoutés à un `JsonObjectBuilder` afin de construire le flux complet.

Il vous faudra modifier le code Javascript afin de prendre en compte la direction.