

# Examen - LI 328

Ludovic Denoyer - Sylvain Lamprier

16 avril 2013

Les documents sont autorisés. Les exercices sont indépendants les uns des autres, mais il est essentiel de lire le **Contexte** de cet examen pour comprendre de quoi il retourne.

## Contexte général

Nous allons nous intéresser ici au développement d'un site Web de type DIGG/REDIT. Un tel site Web est basé sur le principe suivant :

- Des utilisateurs peuvent s'inscrire (en utilisant le nom+prénom) et avoir des amis (identifiés par leur login)
- Un utilisateur peut **ajouter une news** au site si elle n'existe pas déjà dans le site, c'est à dire référencer une page Web (une URL) pour signaler son existence aux autres utilisateurs. Chaque URL est identifiée par un identifiant (un nombre hexadécimal de 32 bits unique pour chaque news)
- Un utilisateur peut **aimer** ou **détester** une news ajoutée précédemment par un utilisateur
- Un utilisateur peut **commenter** une news ajoutée précédemment par un utilisateur

## Exercice I - Spécifications des Web Services (3 points)

Pour chaque service, vous spécifierez les entrées du service, et donnerait un exemple de sortie au format JSON. Par exemple :

```
== Service Login
== Entrée:
  login : le login de l'utilisateur
  password : le mot de passe (en clair...) de l'utilisateur
== Description:
  Le service permet de s'authentifier sur le site et renvoie, en cas de login réussi,
  une clef de session. En cas d'erreur, le service renvoie un message d'erreur
== Erreurs possibles:
  * Login inconnu
  * Mot de passe incorrect
== Exemple de sortie:
  {key = "0BAE569F3AC"} - si login OK
  {error = "Login inconnu"} - en cas d'erreur
```

### Question 1

**Service de Création de compte** :- ATTENTION : si la création de compte est réussie, le nouvel utilisateur est automatiquement loggué dans le site.

### Question 2

**Service Like/Dislike** j'aime/je déteste - en un seul service.

### Question 3

**Service d'informations sur une news** : permet de récupérer les informations sur une news en fonction de son identifiant. Les informations contiennent l'URL, l'utilisateur qui a rajouté cet news, le nombre de *j'aime*, le nombre de *je déteste*, la date d'ajout et le titre de la news.

### Question 4

Quels autres services proposeriez vous que le site soit complet (une à deux lignes de description par service) ?

## Exercice II - Bases de Données (3 points)

On s'intéresse au stockage des différentes données du site. Nous utiliserons **uniquement** une base de donnée MongoDB pour stocker les informations de ce site.

### Question 1

Rappelez l'intérêt d'utiliser une base MongoDB/NoSQL plutôt qu'une base de type SQL.

### Question 2

On considère qu'une news décrite par : son identifiant (32 caractères), son URL, l'utilisateur qui l'a ajoutée, la date d'ajout et le titre de la news. On souhaiterait stocker **dans le même enregistrement** les j'aime/je déteste ainsi que les commentaires. Soit la news suivante :

```
Titre: Euro 2012 - Yanga-Mbiwa et Gourcuff écartés
Auteur de l'ajout: ldenoyer
Date de l'ajout: mar, 29 mai 11:23:00 2012
URL: http://fr.sports.yahoo.com/29052012
```

```
J'aime par slamplier, fhollande
Je déteste par nsarkozy
```

Commentaires:

- \* Date: mar, 29 mai 12:32:00 2012
- \* Auteur : ygourcuf
- \* Commentaire: pas terrible cette sélection

Proposez un format de stockage en JSON de cette news.

### Question 3

Si l'on considère que les news sont stockées dans une collection **examen** sur une BD **li351** du serveur mongodb **li351.com**, écrivez la fonction JAVA permettant de stocker une news en fonction de son URL, son titre, la date d'ajout (sous forme de chaîne de caractères), son auteur.

### Question 4

Ecrivez la fonction JAVA permettant d'afficher le titre des news que *ldenoyer* aime.

## Exercice III - Servlets (5 points)

### Question 1

Rappelez le principe des servlets. Quels sont les différentes possibilités de réponse que l'on peut envisager lorsque l'on travaille avec des Servlets. Quels sont les avantages d'une réponse au format JSON ?

### Question 2

Ecrivez une servlet qui prend deux nombres en paramètres, et retourne la multiplication de ces deux nombres (format JSON).

### Question 3

Ecrivez une servlet permettant d'implémenter le service de Login décrit dans l'exercice 1. Vous considérerez que les utilisateurs sont stockés dans la base MongoDB sous un format **que vous spécifierez**. Notez par ailleurs que la clé fournie pour la session doit être temporaire (elle doit donc posséder une date de péremption).

## Exercice IV - Map/Reduce (3 points)

Si l'on considère maintenant que les j'aime/je déteste sont stockés dans une collection différente de celle des news sous la forme :

```
{
  id_news = "A09B8E8CFF...",
  login = "ldenoyer",
  score = "1"
}
```

avec  $score = -1$  si l'utilisateur déteste.

### Question 1

Ecrivez les fonctions map et reduce permettant de calculer le score final (la somme des scores) de toutes les news. Par exemple :

```
{news= "A09B8E8CFF...", score_total = 38}
{news= "0BAE569F3AC...", score_total = -25}
...
```

### Question 2

Ecrivez les fonctions map et reduce permettant de calculer le score moyen donné par chaque utilisateur sur toutes les news - on considèrera pour un utilisateur donné, uniquement les news scorées. Par exemple :

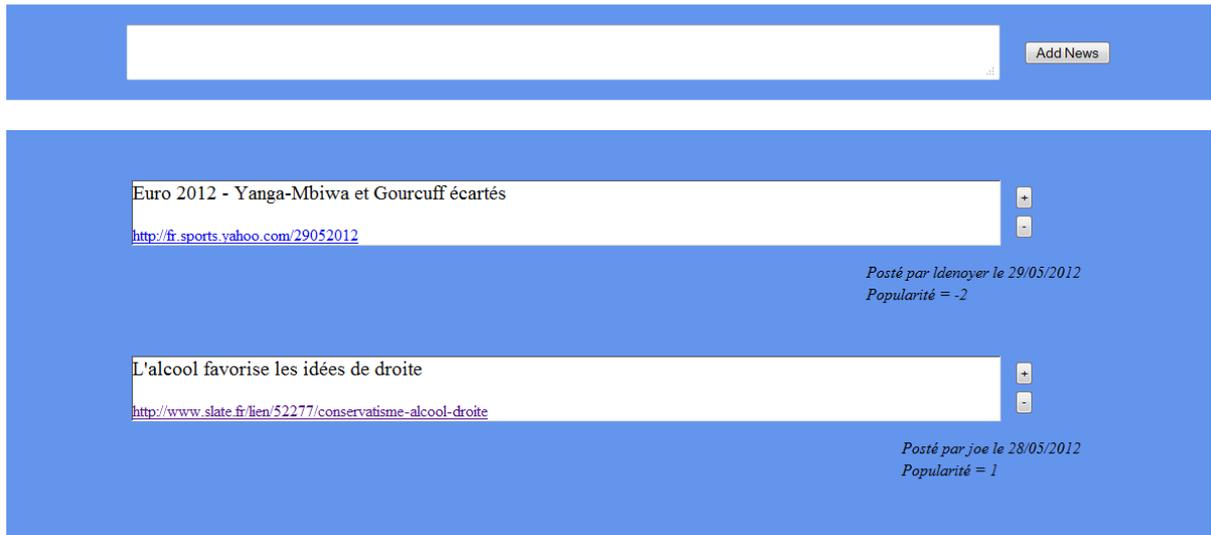
```
{login= "ldenoyer", score_moyen = 2.3}
{login= "slamprier", score_total = -5.3}
...
```

## Exercice V - Développement d'un client (6 points)

On s'intéresse ici au développement d'un client simple, permettant aux utilisateurs d'ajouter et de noter des news.

### Question 1

Proposez le code HTML/CSS permettant d'obtenir une interface semblable à celle donnée par la figure suivante (on affichera ici des commentaires fictifs, donnés en "dur" dans le code HTML) :



## Question 2

On souhaite maintenant faire en sorte que les news affichées correspondent aux dix dernières news ajoutées sur le site. Pour cela, on suppose que le serveur offre le service suivant :

== Service LastNews

== Entrée:

nb : nombre de news

== Description:

Le service retourne les nb dernières news postées sur le site.

En cas d'erreur, le service renvoie un message d'erreur

== Erreurs possibles:

\* Probleme serveur

== Exemple de sortie:

```
{news =
  [
    {id="0BAE569F3AC", titre="Euro 2012 - Yanga-Mbiwa et Gourcuff écartés",
      url="http://fr.sports.yahoo.com/29052012",
      auteur= "ldenoyer", date= "29/05/2012", score=-2},
    {id="AA425F89ABB", titre="L'alcool favorise les idées de droite",
      url="http://www.slate.fr/lien/52277/conservatisme-alcool-droite",
      auteur= "joe", date= "28/05/2012", score=1}
  ]
} - si Ok (correspond à la liste de news affichée dans la figure)
```

```
{error = "Probleme serveur"} - en cas d'erreur
```

En considérant que les communications doivent se faire en asynchrone par le biais d'AJAX et que la servlet correspondant aux service LastNews est stockée sur le même domaine que le code client, écrire le code javascript permettant de récupérer les dix dernières new ajoutées sur le site (uniquement le code de l'appel AJAX). En cas de succès de la communication, on appellera une fonction traiteReponseNews (à écrire dans les questions suivantes) qui attend la réponse au format texte. En cas d'échec on affichera un message "Echec de la communication avec le serveur".

## Question 3

Écrire une classe Javascript News qui vise à regrouper les informations d'une news et propose une fonction getHtml() qui retourne le code HTML de la news (on pensera à faire en sorte

que le clic sur les boutons + et - appellent une fonction `modifScore`, que l'on suppose déjà écrite, à laquelle on passe l'identifiant de la news et la valeur du bouton cliqué en arguments)

**Question 4**

Écrire une classe Javascript `ListeNews` qui correspond à une liste d'objets `News` et propose une fonction `getHtml()` qui retourne le code HTML de la liste de news

**Question 5**

Rappeler pourquoi il est préférable d'utiliser la fonction `JSON.parse` plutôt que la fonction native `eval` pour reconstruire un objet réponse à partir d'un texte au format JSON

**Question 6**

Écrire une fonction `ListeNews.revival` qui définit la manière de reconstruire un objet `ListeNews` à partir d'un texte au format JSON. Cette fonction sera utilisée dans la fonction `traiteReponseNews` de la manière suivante :

```
var liste=JSON.parse(json_texte,ListeNews.revival);
```

avec `json_texte` correspondant au texte retourné par la servlet `LastNews`

**Question 7**

Écrire la fonction `traiteReponseNews` permettant d'afficher la liste de news retournée par le serveur dans l'interface client