

## MIF 13 – Programmation Web – Examen

Durée : 1 heure 30 – Documents autorisés – Ordinateurs, calculatrices et téléphones portables interdits

### Questions de cours (barème : 10 points)

1. Côté serveur, à quoi correspond la partie « Vue » du modèle MVC lors du traitement d'une requête AJAX ? Quel type d'objet correspond le mieux à son implémentation ?
2. Quelle est la différence fondamentale de fonctionnement entre une redirection HTTP et l'utilisation de la méthode `forward()` d'un `RequestDispatcher` ?
3. À quoi cela sert-il d'écrire du JSON dans la partie droite de l'affectation d'une variable en JS ?
4. Si vous deviez utiliser en même temps GWT et jQuery dans une application, à quel problème vous heurteriez-vous ? Citez un exemple de fonctionnalité qui fait doublon entre ces deux outils.
5. En quoi peut-on dire qu'une fonction de callback semble se dérouler dans un thread séparé de celui qui l'a appelée ?

### Étude de cas (barème : 11 points)

Comme chaque année, l'UCBL propose à ses usagers un service de cartes de vœux électroniques. Cette année, c'est vous qui êtes chargé(e) de concevoir cette application Web. Cette application fonctionne de la façon suivante : un utilisateur se connecte à l'application par le CAS. Une fois authentifié, il arrive sur la page d'accueil. De là, il peut visualiser l'aspect général de la carte de vœux et a accès à un lien lui permettant d'envoyer cette carte par messagerie électronique. S'il clique sur ce lien, il accède à un formulaire qui possède les champs suivants :

- Adresse de l'expéditeur (déjà remplie en fonction de l'adresse mail de l'utilisateur dans l'annuaire LDAP et non modifiable)
- Adresse du destinataire
- Titre de la carte
- Texte de la carte
- Choix de la langue de la carte (« français » ou « anglais »)
- Case à cocher pour recevoir ou non un message de confirmation
- Bouton envoyer

S'il clique sur le bouton envoyer, la carte est envoyée à l'adresse indiquée et une page de confirmation / erreur d'envoi s'affiche. S'il a demandé une confirmation, il reçoit un message électronique lui indiquant le résultat de cette opération.

Pour réaliser cette application, vous disposez des composants suivants :

- Une classe `Carte` qui ne possède qu'une méthode statique nommée `getHtml(String titre, String texte, String langue)`. Cette méthode génère le code HTML de la carte de vœux à envoyer.
- Une classe `Message` possédant les champs :
  - o nom de l'expéditeur
  - o adresse du destinataire
  - o titre du message (peut être le titre de la carte, mais pas nécessairement)
  - o texte (ou code HTML) du messageLa classe `Message` contient également une méthode `envoyer()` qui s'occupe de l'envoi par messagerie électronique.
- Une classe `User` qui représente l'utilisateur authentifié. Cette classe permet d'accéder à l'ensemble des champs de l'annuaire LDAP pour l'utilisateur. Pour l'instancier, il suffit d'appeler son constructeur, avec en paramètre le login de l'utilisateur. Les champs sont ensuite accessibles à l'aide d'accesseurs (« getters »).

Cette application ne s'appuiera pas sur un framework et mettra en œuvre un pattern Web MVC push-based.

### Conception (barème : 4 points)

6. Listez les différents éléments nécessaires au fonctionnement de votre application, en indiquant leur nature : pages statiques, scripts, servlets, JSP, JavaBeans (précisez le scope)...
7. Décrivez, à l'aide de diagrammes UML appropriés, leurs communications lorsqu'un utilisateur authentifié clique sur le bouton de validation du formulaire, à partir de la réception des requêtes du client. Les éléments du SI externes à votre application seront modélisés comme des boîtes noires.

### Structure de l'application (barème : 8 points)

8. Écrivez le contrôleur de cette application.

### Programmation (barème : 3 points)

Vous allez maintenant rajouter une partie dynamique à cette application : lorsque l'utilisateur tape des caractères dans le champ destinataire, l'application lui suggère des personnes parmi ses contacts. Pour cela, vous disposez d'un composant côté serveur qui interroge la liste de contacts de l'utilisateur et répond aux requêtes GET sur l'URL « `getContacts?start=xxx` », en renvoyant les contacts qui commencent par la valeur du paramètre « `start` ». Vous n'avez pas à vous préoccuper de son implémentation.

La réponse est un document XML dont la DTD est :

```
< !ELEMENT liste (contact)*>
< !ELEMENT contact (#PCDATA)>
< !ATTLIST contact adresse CDATA #REQUIRED>
```

9. Écrivez la fonction JavaScript de `callBack` qui :

- affiche l'ensemble des contacts suggérés dans un div dont l'ID est « `suggestions` » et rend ce div visible
- fait en sorte que chacun de ces contacts soient cliquables et que lorsque l'utilisateur clique sur l'un d'eux, le div « `suggestions` » devienne invisible, et l'adresse de messagerie du contact soit ajoutée au champ d'ID « `destinataire` ».