

MIF 13 – Programmation Web – Examen

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

Questions de cours (barème : 10 points)

1. Pourquoi ne doit-on pas spécifier l'attribut « width » sur une image en XHTML Strict ? Comment faire dans ce cas pour spécifier la taille de l'image ?
2. Quelle est la différence principale entre les méthodes HTTP « GET » et « POST » ? Dans quels cas choisir un envoi par POST ?
3. Citez deux exemples de cas d'utilisation de réponses HTTP 1.1 dont l'encodage est de type « chunked ».
4. À quoi sert une fonction de callback en JavaScript ? Citez un exemple d'utilisation.
5. Dans une application Web, une page doit répondre à une requête asynchrone du client en renvoyant un document XML (et non XHTML). Vaut-il mieux l'implémenter sous forme de servlet ou de JSP ? Justifiez votre réponse.

Étude de cas (barème : 10 points)

Vous devez concevoir une application Web de partage de photos et de commentaires sur ces photos. Les photos ne pourront être publiées que par les utilisateurs authentifiés mais seront visibles par n'importe quel internaute. Toutefois, seuls les utilisateurs authentifiés pourront récupérer les photos avec leur taille d'origine ; les autres utilisateurs n'auront accès qu'à une version limitée en taille des photos. Les commentaires relatifs à des photos ne seront accessibles et modifiables que pour les utilisateurs authentifiés.

Pour concevoir cette application, vous disposez des éléments suivants :

- un serveur d'authentification CAS et un annuaire de comptes utilisateurs LDAP
- un SGBD contenant les photos et les commentaires des utilisateurs
- une classe Java nommée « Photo », accessible côté serveur et permettant d'encapsuler le contenu d'une photo et de fournir les accesseurs qui font le lien avec la BD (getId, setId, getImage, setImage, getText et setText).
- une classe Java nommée « Resizer », accessible côté serveur et permettant le redimensionnement des photos ; elle possède la méthode :

```
public static Photo resize(Photo source);
```

L'application demandée ne s'appuiera pas sur un framework et permettra les fonctionnalités ci-dessous :

Scénario 1 : upload d'une photo et saisie d'un commentaire

- Connexion d'un utilisateur par redirection sur le CAS
- Création d'un nouveau document photo
 - o Upload de la photo
 - o Saisie éventuel d'un texte de commentaire
- Validation :
 - o Si erreur dans le formulaire : réaffichage des éléments envoyés et d'un message d'erreur
 - o Si pas d'erreur : redirection vers la page de visualisation de la photo, comprenant l'id de la photo dans l'URL
- Déconnexion de l'utilisateur

Scénario 2 : consultation d'une photo et saisie d'un nouveau commentaire

- Interrogation du serveur à l'aide d'une URL contenant l'id d'une photo
- Accès à la photo en mode dégradé et au bouton « Connexion »
- Clic sur le bouton de connexion et redirection sur le CAS
- Retour sur la page de la photo :
 - o Visualisation du texte de commentaire de la photo et du bouton « modifier »
 - o Si clic sur le bouton « modifier », accès à un éditeur WYSIWYG permettant de modifier le commentaire.

Structure de l'application (barème : 6 points)

6. Listez les éléments (pages statiques, scripts, servlets, JSP, JavaBeans...) nécessaires au fonctionnement de votre application.
7. Décrivez, à l'aide d'un diagramme UML approprié, leurs communications lors du scénario 1, à partir de la réception des requêtes du client. Les éléments du SI externes à votre application seront modélisés comme des boîtes noires (CAS, LDAP, Resize).

Programmation (barème : 4 points)

8. Écrivez la JSP qui renvoie une page Web contenant une photo et éventuellement un commentaire, comme indiqué dans le scénario 2. Attention, son comportement ne sera pas le même si l'utilisateur est authentifié ou non.