

10. Quel est le problème principal d'un dialogue applet-servlet dans un contexte d'e-commerce où un client réalise plusieurs requêtes ? Et comment est-il résolu explicitement (cookies, réécriture d'URL) ou de manière plus abstraite (session) ?

Le protocole HTTP est un protocole sans état. Une fois la transaction terminée, aucune information ne permet d'identifier le client lors de transactions futures. On ne peut donc pas travailler dans un contexte de « Session ». Ce qui peut être gênant pour un caddie virtuel par exemple où il faut absolument reconnaître l'acheteur de chacun des achats.

Solutions au protocole HTTP :

- La réécriture d'url : cette technique a pour principe de compléter l'URL de la servlet d'une information complémentaire que l'on désire transmettre comme par exemple dans ce cas ci un identificateur de session. Cette information se trouve entre le path et la query-string c'est-à-dire entre l'URL proprement dite et la chaîne des paramètres du formulaire commençant par « ? ». Elle est récupérable au moyen de la méthode de HttpServletRequest :
 - `public abstract String getPathInfo()`
- Un Cookie, est une information (plus précisément un fichier texte) envoyée à un browser par un serveur qui vient d'être accédé afin que cette information soit sauvegardée sur le client. Chaque fois que le browser s'adressera à ce serveur, il lui enverra le contenu du cookie. De cette manière, un serveur peut dans le cas qui nous occupe ici identifier d'une manière unique un client. Il existe une classe Cookie dont le constructeur est :
 - `public Cookie(String name, String value).`

Les paramètres HttpServletRequest et HttpServletResponse possèdent des méthodes de lecture et d'écriture des cookies :

- `public abstract void addCookie(Cookie cookie)`
- `public abstract Cookie[] getCookie()`

Cette dernière renvoie un tableau contenant tous les cookies fournis par le browser dans sa requête sous la forme d'un tableau. Pour tout cookie existant, on peut récupérer son nom et sa valeur par les méthodes de la classe Cookie :

- `public String getName() ;`
- `public String getValue() ;`

- Le suivi de session, le JSDK 2.0 fournit un « API Session Tracking » qui permet de gérer une session http au moyen d'un interface HttpSession. Puisque HttpSession n'est qu'un interface, l'objet qui implémente celui-ci provient de HttpServletRequest, on y trouve donc la méthode :
 - `public abstract HttpSession getSession(boolean create)`

Le rôle de l'objet session est de conserver des informations concernant l'interaction entre le serveur et le client. Ces informations peuvent être placées dans l'objet session par les méthodes de HttpSession :

- `public void putAttribute(String name, Object value)`

Et récupéré au moyen de :

- `public Object getAttribute(String name)`

Une session peut expirer soit automatiquement après un certain temps d'inactivité dont la valeur est fixée par le serveur ou alors explicitement par l'utilisation de la méthode :

- `public abstract void invalidate()`