Cookies HTTP

Alexandre Niveau

GREYC — Université de Caen

En partie adapté du cours de Jean-Marc Lecarpentier

Le problème d'HTTP

- HTTP est un protocole *sans état* :
  - ne garde aucune trace des requêtes faites au serveur
  - exécute chaque requête indépendamment des autres
- impossible de conserver le contexte en cours !

Conservation du contexte

- Problème : entre deux pages d'un même site, on a souvent besoin de conserver le contexte, c'est-à-dire des informations sur l'internaute et ses actions
- Exemples :
  - identification de l'internaute
  - statut de l'internaute (visiteur, administrateur, etc.)
  - son parcours dans le site
  - les choix effectués (panier d'achats, préférences, etc.)

Les solutions

- Utiliser les paramètres d'URL pour passer les informations : limité en taille, pas très agréable pour l'internaute, valeurs visibles dans la barre d'adresse, pas propre
- Utiliser des champs cachés de formulaire POST sur chaque page : alourdit le code HTML (et donc le poids du fichier à transférer), et ne marche pas si l'internaute quitte le site puis revient
- Ces problèmes sont réglés par l'utilisation de *cookies* :
  - le serveur stocke des données chez le client dans un fichier appelé cookie
  - à chaque requête vers ce serveur, le client transmet les données dans l'en-tête HTTP
  - les cookies peuvent survivre à la déconnexion au site
- **Attention** : aucune de ces techniques n'empêche l'internaute de modifier les données envoyées au serveur !

Fonctionnement des cookies

- Les cookies HTTP ont été créés en 1994, pour implémenter un panier d'achats
- Ajoutés officiellement à HTTP en 1997 (RFC 2109 [http://tools.ietf.org/html/rfc2109])

- Exemple de fonctionnement (adapté de Wikipédia [http://en.wikipedia.org/wiki/HTTP_cookie#Implementation]) :
  Le client demande une page au serveur

```
GET /index.html HTTP/1.1
Host: www.example.org
```

Réponse du serveur

- Le serveur lui envoie la page, en demandant au navigateur de créer un cookie :

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: name=value
Set-Cookie: name2=value2; Expires=Wed, 09 Jun 2021 10:18:14 GMT

<!DOCTYPE html>
<html lang=fr>
(suite de la page…)
```

- Le navigateur stocke les informations dans un nouveau cookie, qui restera valable jusqu'à la date demandée

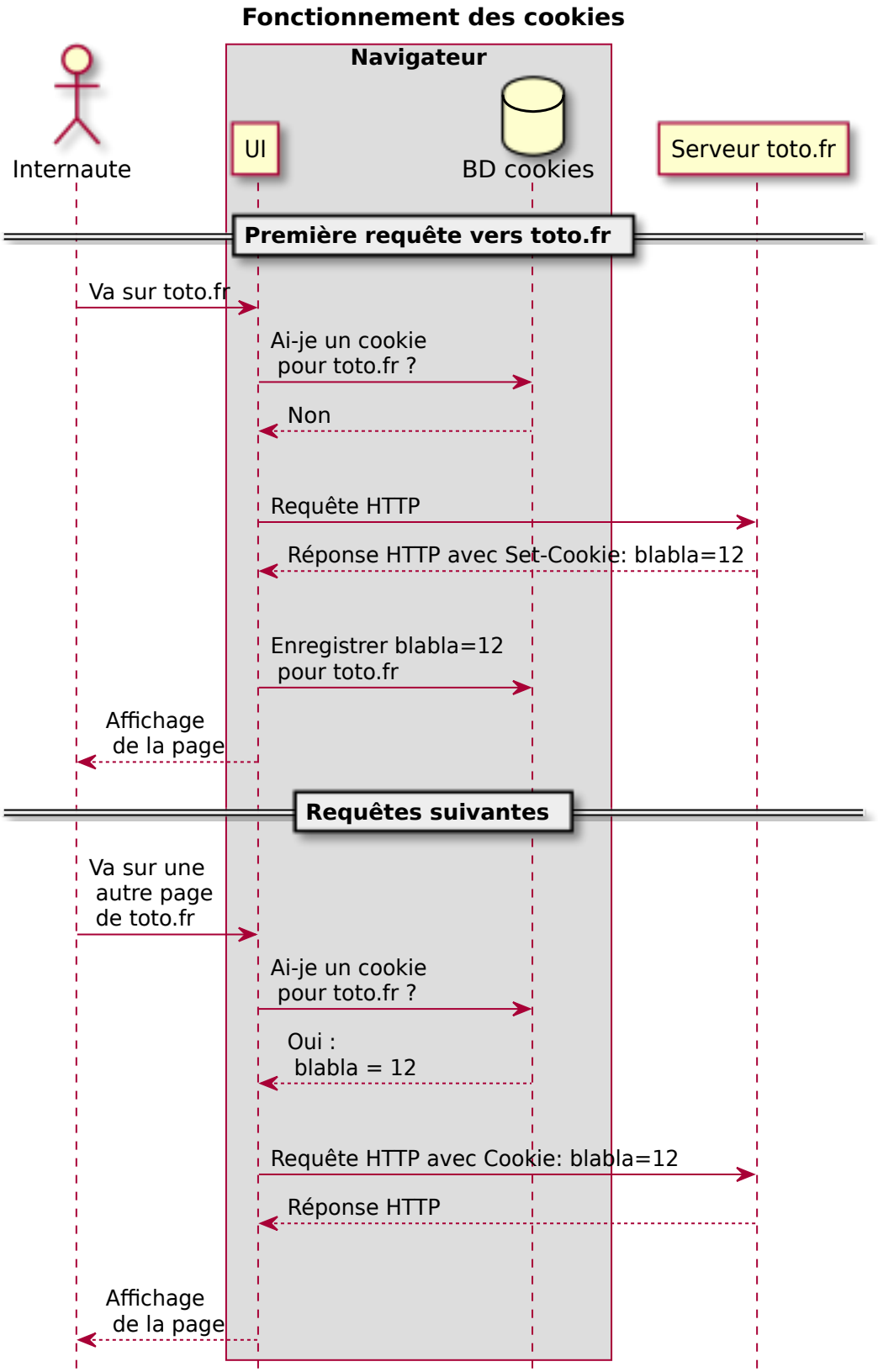
Nouvelle requête du client

- Chaque nouvelle requête du client vers ce serveur sera accompagnée du contenu du cookie :

```
GET /toto.html HTTP/1.1
Host: www.example.org
Cookie: name=value; name2=value2
```

- Le serveur peut modifier une valeur, toujours avec `Set-Cookie`

Diagramme de séquence



Fonctionnement des cookies (lien vers l'image SVG) [img/seq_cookies.svg] (lien vers l'image PNG) [img/seq_cookies.png]

Cookies avec PHP

- PHP possède des fonctions permettant de manipuler les cookies
- Ces fonctions vont s'occuper de modifier les en-têtes HTTP des réponses pour ajouter `Set-Cookie`, et de lire les informations du champ `Cookie` dans les requêtes
- Pour créer un cookie : `setcookie` (voir manuel) [http://fr2.php.net/manual/fr/

function.setcookie.php]

```
setcookie('prenom', 'Toto', time() + (86400 * 7));
```

crée un cookie valable 7 jours et contenant l'information `prenom=Toto`.

- Les cookies envoyés par le client sont accessibles dans le tableau `$_COOKIE` :

```
if (key_exists('prenom', $_COOKIE))
  echo "Bienvenue, " . htmlspecialchars($_COOKIE['prenom']) . " !";
```

- Ne jamais supposer qu'un cookie existe : les navigateurs peuvent les refuser, les clients peuvent les supprimer…
- Attention aux caractères interdits dans les noms de cookie : **espaces**, guillemets, virgule, point-virgule, et antislash. Peut générer des bugs difficiles à trouver.

Limites des cookies

- Les cookies sont plus pratiques que l'utilisation de paramètres d'URL ou de champs cachés, mais pas si différents
- En particulier, l'internaute peut les modifier à loisir
- Solution : n'utiliser les cookies que pour *identifier* l'internaute (par exemple en lui associant un numéro), et ne stocker les informations que côté serveur
- On verra plus tard que PHP propose un mécanisme de *sessions* pour faciliter cela

Spécifications et normes

- RFC 6265 — HTTP State Management Mechanism [http://tools.ietf.org/html/rfc6265] (référence sur les cookies HTTP)

Références et guides

- Manuel PHP sur les cookies [http://fr2.php.net/manual/fr/features.cookies.php]

Lectures complémentaires

- HTTP cookie [http://en.wikipedia.org/wiki/HTTP_cookie] sur en.wikipedia